AD NUMBER
AD902534
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; FEB 1971. Other requests shall be referred to Air Force Armament Laboratory, Attn: DLYD, Eglin AFB, FL 32542.**
AUTHORITY
AFATL per ltr, 7 Sep 1976

AD902534

DDC FILE COPY

AFATL-TR-71-20
VOLUME II

WEAPON SYSTEM EFFECTIVENESS ANALYSIS, OPTIMIZATION AND SIMULATION-PHASE I

VOLUME II. COST EFFECTIVENESS ANALYSIS OF RESOURCE ALLOCATION FOR NETWORK INTERDICTION

COLLEGE OF ENGINEERING
LOUISIANA STATE UNIVERSITY

TECHNICAL REPORT AFATL-TR-71-20, VOLUME II

FEBRUARY 1971

Distribution limited to U. S. Government agencies only; this report documents test and evaluation; distribution limitation applied February 1971. Other requests for this document must be referred to the Air Force Armament Laboratory (DLYD), Eglin Air Force Base, Florida 32542.

AIR FORCE ARMAMENT LABORATORY
AIR FORCE SYSTEMS COMMAND • UNITED STATES AIR FORCE
EGLIN AIR FORCE BASE, FLORIDA

Weapon System Effectiveness Analysis, Optimization And Simulation-Phase I

Volume II. Cost Effectiveness Analysis Of Resource Allocation For Network Interdiction

Lawrence Mann
John Heard Ristroph

FOREWORD

This final report documents the results of research and analysis performed during the period 1 February 1970 to 30 September 1970 by a project team comprised of eight faculty members and six graduate students from Louisiana State University, College of Engineering, Baton Rouge, Louisiana under contract F08635- 70-C-0053 with the Air Force Armament Laboratory, Eglin Air Force Base, Florida. Major Frank A. Roescher (DLYD) was program monitor for the Armament Laboratory. Dr. Adrain E. Johnson, Jr., Department of Chemical Engineering, was project director for Louisiana State University.

The work undertaken by the project team consisted of several major and minor tasks dealing with the analysis of weapons systems, each of which was assigned to a small group, typically consisting of one faculty member and one graduate student. Seven of the major tasks were completed during this phase of the program and are included in this report as Volumes I through VII. Five additional major tasks are currently underway, and the results will be included in a separate technical report upon completion of Phase II of the contract.

This technical report has been reviewed and is approved.

Thomas P. Christie
THOMAS P. CHRISTIE
Chief, Analysis Division

ABSTRACT

A cost-effectiveness analysis was performed on the problem of resource allocation for supply network interdiction. Subject to availability constraints, Air Force weapon systems are allocated to targets within a supply network. The measures of system performance are cost in dollars and effectiveness in pounds of supplies interdicted. By means of a limited enumerative approach, solutions may be found such that no other alternative assignments of aircrafts to targets exhibit simultaneously a higher effectiveness but lower cost than the solutions. Data elements describing the network and weapon system characteristics and weapon system availabilities are processed by a computer model in such a manner as to provide information which will aid a field commander's decision processes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A_i is the expected percent of a target destroyed by the i^{th} weapon system which attacks it.

$A_{i,j}$ is the percent damage expected to be inflicted on a target during the i^{th} attack by weapon system type "j", when all weapon systems are attacking the same target.

$A^j_{i,r}$ is the probability of target "r" being killed by weapon system "i" after the target has already been attacked "j - 1" times.

B_i is the probability of kill of the i^{th} weapon system by a target if that weapon system were to attack first.

$B_{i,j}$ is the percent damage expected to be inflicted on weapon system type "j" during the i^{th} attack when all weapon systems are attacking the same target.

$B^j_{i,r}$ is the probability of weapon system type "i" being killed by target "r" after the target has been attacked "j - 1" times.

C is the total cost, in dollars, incurred in performing some set of attacks. This includes ordnance, fuel, routine maintenance, and attrition costs.

C_j is the cost in dollars, of the plane type and pilot used by the i^{th} weapon system which attacks the same target.

C^N_M is the number of unique combinations which may be obtained by choosing M elements at a time from a set of N elements.

$D_{i,r}$ is the number of type "i" weapon systems, less than or equal to the total availability of weapon system type "i", which have been deleted from consideration with regard to an attack on target "r".

E is the total effectiveness, in pounds, of some set of attacks.

$E^j_{i,r}$ is the effectiveness, in pounds, of an attack by weapon system type "j" on target "r" after it has been attacked "j - 1" times.

F_i is the i^{th} element of some set of elements.

H is a set of elements.

H_k is the k^{th} subset of H.

$H_{k,p}$ is the p^{th} element of the subset H_k.

i	is used as a subscript. It may be any natural number.
j	is used as a subscript. It may be any natural number.
k	is used as a subscript. It may be any natural number.
M	is the number of elements to be chosen at a time from some set.
N	is the total number of elements in some set.
N_i	is the number of the i^{th} type of weapon system which is available.
O_i	is the ordnance, fuel, and routine maintenance cost, in dollars, for weapon system type "i" when only one target is being attacked.
p	is used as a subscript. It may be any natural number.
P_i	is the probability that the i^{th} weapon system which attacks the same target will be destroyed.
r	is used as a subscript. It may be any natural number.
S_i	is the state of a target immediately before it has been attacked for the i^{th} time.
$T^j_{i,r}$	is the attrition, fuel, ordnance, and routine maintenance cost, in dollars, of an attack on target "r" by weapon system type "i" after that target has been previously attacked "j - 1" times.
x	is any number greater than or equal to 0 and less than or equal to 1.
y	is used as a subscript. It may be any natural number.
z	is used as a subscript. It may be any natural number.
/ /	is used to define a vector function which is given in Equation (7).
$\succ$	is used in the ranking of vectors; refer to Equation (9).
$>$	means "is greater than."
$\geq$	means "is greater than or equal to."
$<$	means "is less then."
$\leq$	means "is less than or equal to."
$=$	means "is equal to."

SECTION I

INTRODUCTION

Background

The objective of this work is to provide a tactical field Air Force commander who has limited resources with a decision-making tool. The decision to which this research is directed involves the problem of determining which aircraft to dispatch against targets in a supply network. His decision uses as inputs types of weapon systems, number of types operational (availability), and network and target parameters. As selection criteria, the mission cost (expected dollar cost of a mission) and effectiveness will be used. Effectiveness is the expected pounds of supplies interdicted for a short period of time. For the purposes of this study, a short period of time is that time period in which enemy supplies may not be effectively rerouted. It is assumed during this time period that those who have control over the target cannot respond by replacing lost or damaged components of the target.

It should be emphasized that, under the system developed, the field commander still retains actual command. This is in contrast to other work done in this field of endeavor which optimizes decisions but removes the field commander from the decision process. That is, in using the algorithm developed here, the printout provides the commander with a series of best alternatives. If these alternatives do not coincide with the commander's judgment as to what resources should be allocated to the mission, then the cost and effectivness of other alternatives is displayed to him to aid in his decision-making process.

Early attempts at the problem involved the search for a criterion which considered cost and effectiveness together. Various ratios and relationships were tried without success. In the program developed, both cost and effectiveness have been built into the printout displays so that those using this information for decisions may evaluate the two together in light of the exigencies of the situation. When all possible combinations of weapon type, quality of types, and target parameters are considered, one or more of the possible alternative system configurations may exceed or may be below acceptable cost and/or effectiveness levels. In these cases it appears prudent to search for these abnormalities as soon as possible, thereby minimizing the number of systems for further consideration. Such a preliminary search is performed.

When this type of problem is considered, it appears wise to approach the solution in a two-step fashion. The first step involves the establishment of feasible criteria and limits so that the search algorithm may not be active over the entire range of possible combinations with regard to the number of aircraft to be sent on a mission.

From the above discussion, the problem then resolves itself into the consideration of two separate concepts. The first concept involves a discussion of the problem and the relation of cost and effectiveness to it. The second concept involves the use and adaption of the principles of combinatorial numbering systems to ascertain the feasible alternatives.

One of the missions of an Air Force is to interdict enemy supplies or to stop them from reaching their destination. Another is to provide close air support for combat troops. This research is concerned with the interdiction problem.

An essential problem in the interdiction mission is that of allocation, i.e., determining which weapon systems attack which targets. Further, the order in which the planes attack is to be considered.

In the interdiction role, the availabilities of planes and weapon systems are a constraint on the problem. Second, the effectiveness of a proposed attack must be determined. Third, the cost of a proposed attack must be evaluated.

The terminology of the problem will now be established.

Terminology

A system may be defined as some combination of components which convert input into output. Any system may, in turn, be comprised of smaller subsystems.

The research problem may be viewed as a system. The data base describing the weapon system availabilities and enemy supply network constitute input. The method by which this data is converted into usable information is the system. The output is the configuration of attacks and their resultant costs and effectivenesses.

This system may be viewed as a subsystem wherein the larger encompassing system is that of an Air Force Tactical Command. Achieving the best resultants from a subsystem must not hinder total system performance.

Optimization is attaining that level of performance of a system which best satisfies the prescribed measures of performance.

System analysis includes the following tasks, listed in the order of performance:

1. The desired output or that which is needed must be determined. Preferably, the outputs are measurable. If not, they must be quantified.
2. The desired inputs or that which is to be converted into the output must next be determined.
3. The relationship of system components which optimally converts

input into outputs must then be determined.

Cost-effectiveness analysis refers to a particular type of systems analysis in which the measures of performance of a system are cost and effectiveness. This is also referred to as cost-benefit analysis.

A supply source is an origin from which supplies are sent.

A supply sink is a destination to which supplies are sent.

A trans-shipment point is a point which acts both as a source and as a sink. An example of this is a dock at which supplies are transferred from incoming rail cars to outgoing ships.

A storage location is a place where supplies are stored. There is a maximum of one supply location per branch.

A branch is a homogeneous route containing either no storage locations or one storage location. The term "homogeneous" indicates that a branch has the same defense characteristics and the same vehicle capacity along its entire length. If the defense characteristics or vehicle capacity change or if there is more than one storage location, the branch must be segmented into new, smaller branches. The term "route" refers to roadways, railways, canals, paths, or similar entities which can support vehicular traffic.

A vehicle is any supply carrier.

A supply network is a system composed of supply sources, branches, trans-shipment points, vehicles, storage locations, and supply sinks.

A plane is an aircraft of a basic body type, for example, a B-58 or a B-52.

A weapon system is a plane having a particular set of attack characteristics and capabilities. For example, a B-58 with 10,000-pound bombs is a different weapon system from a B-58 with [illegible],000-pound bombs.

An attack is the act of a specified weapon system engaging a particular target.

Attack combinations are acts of more than one weapon system engaging one or more targets. The attacks do not occur in any specified order.

An attack permutation is an attack combination in which the attacks occur in a specified order.

Target state is defined as the pounds of supplies which can be interdicted by completely destroying that target. For example, if

a storage location contains seventy pounds of supplies, then the state of that target is seventy.

Weighting refers to multiplying a target state by some dimensionless number so that the target states have, as their common measure, their utility to the enemy. For example, if rifles were worth fifty times what bread is worth, then the target state of the target containing rifles would be multiplied by fifty.

Short run refers to an interdiction problem in which the time period is so short as not to allow the supplies to be rerouted to another location closer in time to a supply sink from the original location at which the attack was made.

A *long run* is any time period longer than that of the short run.

Statement of the Problem

It is the purpose of this research to design a system to perform a cost-effectiveness analysis on the problem of short-run enemy supply network interdiction by some friendly Air Force weapon systems.

The result of the analysis was the development of a computer model capable of converting data describing the problem into information which is designed to aid a field commander in his decision processes.

SECTION II

LITERATURE BACKGROUND

In the literature survey, the methodologies of cost-effectiveness analysis and various optimization techniques were researched. Additionally, other research bearing on the problem was investigated and the relevant literature is abstracted.

The basic premise of a cost-effectiveness (C-E) analysis is that resources are a real constraint in relation to all of their possible uses[5]. With unlimited resources, it would not be necessary to optimally allocate them. C-E's basic purpose is to indicate, usually in dollars, the cost associated with different alternatives and conditions whose effectiveness is determined by some method. Thus, it is a decision aid.

C-E analysis may be described by the following procedures. These procedures reflect a combination of the work of six authors [1],[4],[6],[7],[8].

1. Define the problem. In doing so, define the measures of cost and effectiveness and the parameters which affect these measures.
2. Determine the alternatives which will satisfy the problem. Delete from consideration those alternatives which are infeasible due to availability constraints.
3. Develop a model which relates the problem parameters to effectiveness and cost in terms of available data. List the nonquantifiable decision parameters.
4. Develop an efficient information storage and retrieval system for effective use of the necessary data.
5. Present the information obtained by the models as a decision aid and implement the desired decision.

The cost-effectiveness criterion may not be stated. For any specified upper level of cost, it is desirable to obtain a maximum of effectiveness. Conversely, for any specified lower level of effectiveness, it is desirable to obtain minimal cost. For example, if alternative A has an equal or higher effectiveness but lower cost than alternative B, then by the cost-effectiveness criterion, alternative B may be deleted from consideration.

This criterion indicates that the ratio of cost to effectiveness (C/E) is not to be used to delete one alternative in deference to another. For example, assume that alternative X has an effectiveness of 5 (pounds of supplies interdicted) and a cost of 3 (dollars), and alternative Y has an effectiveness of 4 (pounds) and cost of 2 (dollars). Using the cost-effectiveness criterion, neither alternative may be deleted from consideration although the C/E ratios of X and Y are 0.6 and 0.5, respectively. Alternative Y has a lower dollar-per-pound ratio than does alternative X, but its effectiveness is one pound less. In using some other alternative, Z, to obtain this unit of

effectiveness, the total cost of alternatives Y and Z may exceed the cost of alternative X. Thus, the C/E ratio is not used as a method for deleting alternative courses of action from consideration.

As the cost and effectiveness values are very often statistically expected values, it is desirable to describe the distribution of values about the expected values. (A statistically expected value is the sum of the products of all possible values multiplied by their respective probabilities of occurrence.)

Hatry[5] reports three ways of doing this:

1. Qualitative descriptions.
2. A graph of cost versus probability that cost will not be exceeded.
3. A graph of cost versus effectiveness in which, instead of a line portraying the relationship, a confidence band is used.

These techniques may be used only when there is knowledge concerning the input distributions describing the system.

For studies such as those described here, there are three types of cost-effectiveness models[5] which may be classified into categories.

The first type is the system configuration studies or system design studies. Here the optimal design of one system and its component characteristics are desired.

The second type is the system comparison study. This is used often, or concurrently with, system design studies to compare two or more systems for the same mission.

In the third type, force structure studies, the blend of systems-over-time and cost-and-effectiveness-levels-over-time are examined.

There are two categories: those which differ in degree and those which differ in kind. A comparison of two defense systems, both employing antiaircraft artillery, would differ in degree. However, these two defense systems would differ in kind from one which employs a sweeping laser beam.

Methods are available for finding the optimal performance level of a system which has a scalar measure of performance and can be described mathematically. A scalar measure may be used when the quality of a system is to be judged only on one measure of performance. For example, in industry, dollars profit is often used as a scalar measure of performance. The problem statement has an objective function expressing system performance as a function of system parameters and has constraint equations or inequalities which specify the range over which the values of the parameters, or combinations thereof, may vary. In most mathematical programming, the parameters are continuous within their specified ranges. For problems which are linear in nature, some algorithms

find integer solutions (integer programming, assignment problems, and transportation problems). Dynamic programming is capable of dealing with discrete nonlinear problems.

For a specified maximal level of cost or for a specified minimal level of effectiveness, the cost-effectiveness problem may sometimes be treated as a mathematical programming problem. However, when neither is specified, mathematical programming may not be used because there is no longer a scalar objective which is required for mathematical programming algorithms.

Mathematical programming also fails for non-linear problems involving discrete alternatives. The interdiction problem studied in this research is such a problem, i.e., either a B-52 or a B-58 may be sent to attack a target, not 0.7 of one and 0.3 of another.

If an upper level of cost were specified for an interdiction problem, then the scalar measure of pounds of supplies interdicted for this cost or less may be used. The interdiction problem would have a scalar measure of performance but discrete courses of action.

Either dynamic programming or a limited enumerative technique may be used to solve this type of problem. Depending upon the particular interdiction problem, either method may be the more efficient with regard to either the time required to obtain a solution or the computer memory storage requirements.

The limited enumerative procedure offers an advantage in that it is not necessary to specify a maximum level of cost. If dynamic programming were used, the solution having the highest level of effectiveness without violating the cost constraint would be obtained. However, there may be another alternative which, although it slightly exceeds the minimal cost, would have a much higher effectiveness. Also, an alternative which has a slightly lower level of effectiveness than the dynamic programming solution but a vastly lower level of cost may be ignored.

If dynamic programming were used, it would be possible to find an optimal interdiction policy. By failing to list the other alternatives, it is possible to ignore alternatives which for a slight increase in cost would have a large increase in effectiveness. This incremental level of effectiveness may reduce battlefield costs tremendously. By using dynamic programming for this problem, it would be possible to optimize the interdiction costs but to suboptimize the entire battle costs.

The limited enumerative approach would yield not only the dynamic programming solution but also the other alternatives. The limited enumerative procedure was the approach used for this research.

SECTION III

PROBLEM CONFIGURATION

The model for the problem of aerial enemy supply network interdiction will now be described. This includes a statement of system inputs which are the data base, the system, and system outputs which are the costs and effectivenesses of the possible attack combinations. The alternative methods of interdiction will be discussed first.

Enemy supplies may be interdicted by several methods, of which the first three are:

1. Destruction of the supplies by aerial weapon systems.
2. Destruction of the vehicles by aerial weapon systems.
3. Destruction of the network branches by aerial weapon systems.

These first three alternatives may also be accomplished by artillery. This is a fourth alternative.

The fifth alternative is to accomplish interdiction by the use of ground troops.

A sixth alternative is an approach to the more general case within which the particular problem studies is imbedded -- that of nullifying the effect of enemy supplies insofar as combat capability is concerned. In this alternative, the supplies are permitted to flow through the network. Then the utility of supplies may be negated by close air support of combat troops.

The scope of this investigation included the first three alternatives and, since they differ in degree, the determination of the results of different mixes of these alternatives. The modeling of the use of artillery could be done in a manner similar to that which is used for the first three alternatives. The fifth and sixth alternatives, which both incorporate ground troops into their models, differ in kind from the first three alternatives and require a different modeling technique.

The first three alternatives are to be examined. The last three are recommended for future research.

System Inputs

The system inputs are the data describing the network and the weapon systems. These are:

1. The network data:
    a. Vehicular flow rates per hour on each branch.
    b. The vehicular loading on each branch, expected pounds per vehicle.

c. The pounds of supplies stored on each branch.
d. The distance in hours from each branch to a supply sink.
e. A weight factor for each target (supplies or vehicles on a specified branch or the branch itself) which reflects the relative importance of the supplies at the target.

2. The attack data:
a. Probability of a specified target being killed by a specified type of attacking weapon system (the percent damage expected).
b. Probability of a specified attacking weapon system being killed by the defenses around a specified target.

3. The cost data:
a. Variable cost (fuel, ordnance, and routine maintenance) for each attack in dollars.
b. Fixed cost (plane, pilot) for each attack in dollars.

4. Availability data:
a. Plane and weapon system availabilities by type.
b. Minimum and maximum number of weapon systems to be sent.

The System

The system which converts the input into output is an algorithm which is executed on any computer which uses Fortran language.

Once the necessary data has been read in, the model performs a search to determine which attack permutations have measures of cost and effectiveness which cannot be deleted using the limits established for the cost-effectiveness criterion.

System Output

After the permutations have been found, they and their costs and effectiveness are printed out. The measure of effectiveness is pounds of supplies interdicted. The cost is the dollar value required to perform the attack permutation.

System Comparisons

It is noteworthy that, if system comparisons were to be made between the first three alternatives and the fourth and the fifth, a common measure of effectiveness and cost is desirable.

All alternatives except the last (which uses close air support) may use the measures of pounds and dollars. If the following were done, then the last alternative, in which ground troops are used, might also use the same measures.

Consider the result of not interdicting XX pounds of supplies. Use this as the measure of effectiveness. As a cost, use the incremental dollars cost required to accomplish one's battle objectives with the aid of the additional close air support required due to the presence of additional enemy supplies.

SECTION IV

METHOD OF SOLUTION

To obtain attack permutations which satisfy the cost-effectiveness criterion, all attack combinations not known to be infeasible by preliminary investigation over the range of the number of aircraft to be sent are determined. Each combination is ordered into permutations which satisfy the cost-effectiveness criterion. For any permutation of attacks, the supply network characteristics (flow rates, pounds of supplies stored, etc.) are updated in accordance with each succeeding attack. After the calculation of that permutation's effectiveness and cost, then the network characteristics are restored to their original values. The values of effectiveness and cost for the permutation are then compared with those of previously evaluated permutations to see if any prior values would cause the new permutation to be deleted using the cost effectiveness criterion. Those values previously calculated are tested to determine if they could be deleted by the new values. Upon deletion, an attack permutation is erased from computer memory storage. Figure 1 is a pictorial representation of this process.

In this section, the following algorithms will be developed:

1. The algorithm by which the combinations are generated.
2. The calculation of effectiveness and cost values.
3. The optimal permutations of a combination.
4. The procedure by which a permutation is deleted.

In order to facilitate understanding of these algorithms, the main program and subroutines of the computer model will be explained; then the derivations of the algorithms will be given.

Computer Program

The main program is used to coordinate the subroutines. No actual computations are performed in the main program; rather, it is used to call the subroutines. The macro-logic flowchart of the computer program is given as Figure 2.

READIN is the first subroutine called. It reads in the problem data and sets up the information storage and retrieval system. The system inputs are the data items read into the computer.

First, the minimum and maximum number of weapon systems to be sent are read in, followed by various information designed for the user's convenience. Then the availabilities of each type of aircraft and weapon system are read in, followed by data describing the branches. The data for each of the contemplated attacks is then put in the computer. This will be shown more precisely in the example given in Section V.

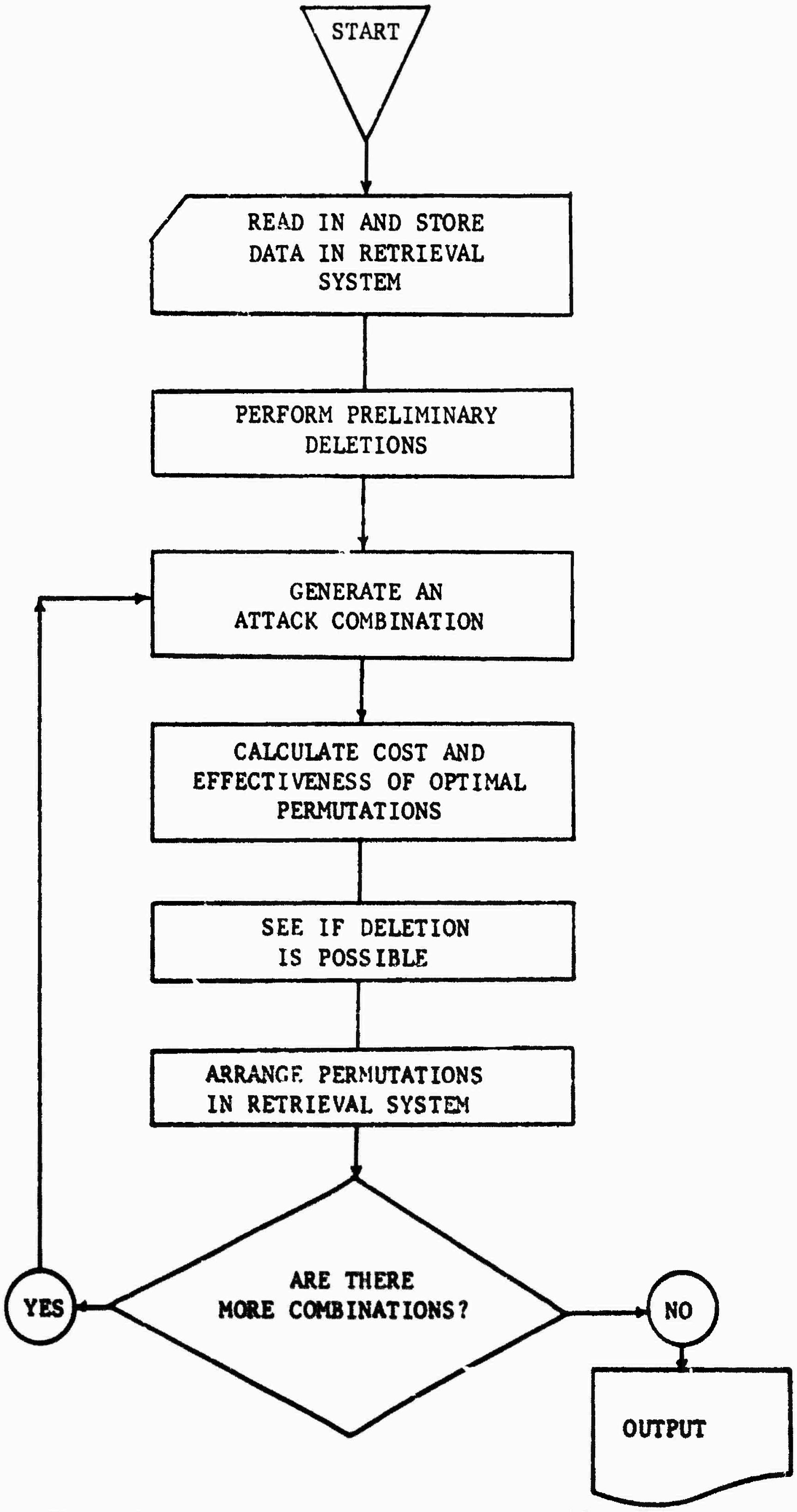


Figure 1. Macro-Logic Flowchart of Combination Algorithm

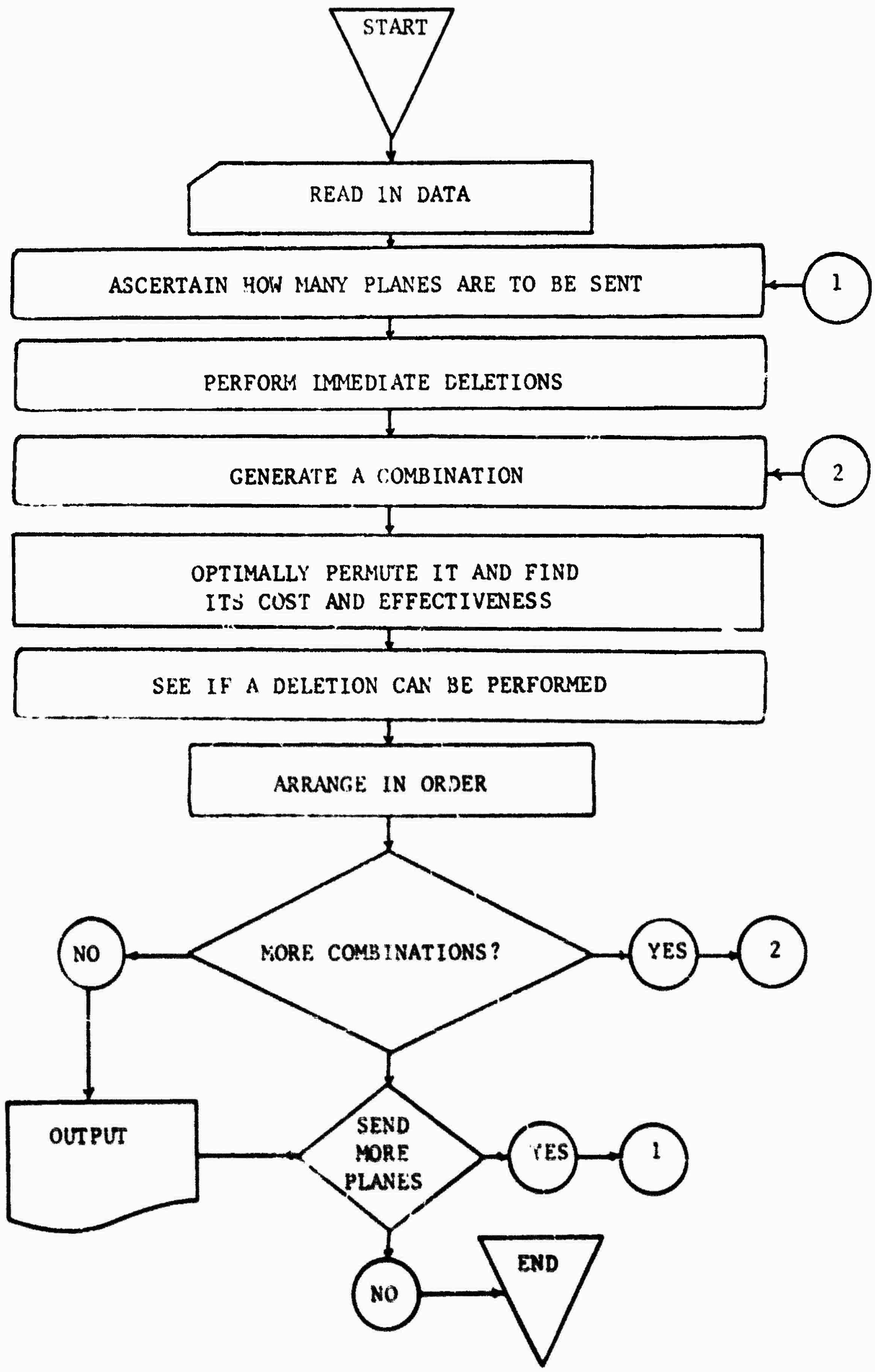


Figure 2. Macro-Logic Flowchart of Computer Model

Any type of numeric identification system may be used to identify the branches and plane and weapon system types. The model converts the user's identification system to its own and then outputs the information in terms of the user's system. As an example, each possible attack for which the data is read into the computer is given a unique attack number. In the printout, this attack number is converted into the weapon system, branch, and target numbers read into the computer by the user.

IMDDLT is the immediate deletion subroutine. After the main program has called READIN, then IMDDLT is called. By scanning the input data, this subroutine immediately deletes from consideration those attacks which are infeasible by using the cost-effectiveness criterion. This is explained later in this section.

COMBO is the next subroutine called. In COMBO, all possible attack combinations are generated by an algorithm presented in Figure 1. After each combination the subroutine EFFEC is called.

EFFEC orders the attack combination into one or more optimal attack permutations by an ordering method. After optimal ordering, the total expected cost and total expected effectiveness of the attack are calculated.

The DELETE subroutine checks to see if either the new attack permutation or any of the old attack permutations may be rendered infeasible by the cost-effectiveness criterion.

Once the feasibility of an attack permutation has been determined, the subroutine ORDER follows. This subroutine arranges either the total effectiveness values in descending order or the total cost values in descending order , or both, as the user desires.

All attack permutations are generated, evaluated, and, if necessary, deleted from that set of permutations generated by sending some fixed number of weapon systems. For example, all permutations obtained by sending two weapon systems are examined, and the information in output and the memory banks of the computer are reinitialized before attack permutations of three attacks are examined.

OUTPUT is used after all permutations of some fixed number of attacks are examined in the manner described above. The resultant information is then output in terms of the user's identification system. This information includes the attack permutations (with each specific attack listed) and the cost and effectiveness values. Summary tables are provided which provide the field commander with total cost and total effectiveness values for the previously listed attack permutations.

Figure 2 shows the macro-logic of the computer model. A computer listing and the micro-logic flowchart of the example problem are shown as Appendices II and III, respectively.

Now that a general explanation of the computer program has been made, the derivations of the computational procedures will be given.

<u>Combination Algorithm</u>

The most fundamental algorithm to the computer model is that which generates the alternative attack combinations. The method of generation will be illustrated prior to giving its proof. The proof that the algorithm generates all possible combinations is somewhat complex.

An illustration of the algorithm is given below. Given the following set of six elements, which may be considered as attacks, all combinations of three elements will be generated. This is the same as choosing all combinations of three weapon systems to be sent at a time at a target when there are six weapon systems available. Call the set "H", its elements "F_i", and the combinations "H_k".

$$H = \langle F_1, F_2, F_3, F_4, F_5, F_6 \rangle$$

$$H_1 = \langle F_1, F_2, F_3 \rangle$$

$$H_2 = \langle F_1, F_2, F_4 \rangle$$

$$H_3 = \langle F_1, F_2, F_5 \rangle$$

$$H_4 = \langle F_1, F_2, F_6 \rangle$$

$$H_5 = \langle F_1, F_3, F_4 \rangle$$

$$H_6 = \langle F_1, F_3, F_5 \rangle$$

$$H_7 = \langle F_1, F_3, F_6 \rangle$$

$$H_8 = \langle F_1, F_4, F_5 \rangle$$

$$H_9 = \langle F_1, F_4, F_6 \rangle$$

$$H_{10} = \langle F_1, F_5, F_6 \rangle$$

$$H_{11} = \langle F_2, F_3, F_4 \rangle$$

$$H_{12} = \langle F_2, F_3, F_5 \rangle$$

$$H_{13} = \langle F_2, F_3, F_6 \rangle$$

$$H_{14} = \langle F_2, F_4, F_5 \rangle$$

$$H_{15} = \langle F_2, F_4, F_6 \rangle$$

$$H_{16} = \langle F_2, F_5, F_6 \rangle$$

$$H_{17} = \langle F_3, F_4, F_5 \rangle$$

$$H_{18} = \langle F_3, F_4, F_6 \rangle$$

$$H_{19} = \langle F_3, F_5, F_6 \rangle$$

$$H_{20} = \langle F_4, F_5, F_6 \rangle$$

From combinational analysis, it is known that the following is an expression for the number of different combinations, C_M^N, which may be formed by taking M items at a time from a set of N items.

$$C_M^N = \frac{N!}{M!\ (N-M)!} \qquad (1)$$

$$C_3^6 = \frac{6 \cdot 5 \cdot 4 \cdot 3!}{3 \cdot 2 \cdot (3)!}$$

$$C_3^6 = 20$$

This confirms that the example gives the correct number of combinations.

The proof consists of the following:

Let the set "H" consist of the integers "1" through "N". These integers may be thought to represent the subscripts of the "F_i" in the example.

$$H = \langle 1, 2, 3, \ldots, (N-1), N \rangle \qquad (2)$$

Let "H_k" represent the k^{th} combination of M items taken from the set "H" where

$$H \geq M > 1. \qquad (3)$$

Represent "H_k" as follows:

$$H_k = \langle H_{k,1}, H_{k,2}, \ldots, H_{k,M} \rangle \qquad (4)$$

where the first subscript of $H_{k,p}$ designates that it is a member of combination H_k, and the second subscript designates its location within what is now considered the vector H_k. For example, $H_{1,2}$ is the second element of the vector H_1. For purpose of the proof, H_k will consist of ordered elements. As the algorithm is sequential in nature, this is necessary. It is to be proven that no two combinations are identical and that all combinations are generated.

For all natural numbers "j" such that

$$1 \leq j < M \qquad (5)$$

require that

$$H_{k,j} < H_{k,j+1}. \tag{6}$$

Induce the following ordering onto the vectors of combinations. Consider the two vectors H_i and H_j. The vector H_i is to be of higher rank than H_j if and only if

$$/H_i/ > /H_j/ \tag{7}$$

where the following is defined:

$$/H_i/ = H_{i,1}\,(10^{M-1})^N + H_{j,2}\,(10^{M-2})^N + \ldots + H_{i,M}(10^{M-M})^N \tag{8}$$

This is written symbolically

$$H_i \rightarrow H_j$$

which indicates that

$$/H_i/ > /H_j/.$$

Theorem A: Given a H_i such that $H_{i,1} < N\text{-}M$, then there may be constructed an H_j such that:

1. $H_j \rightarrow H_i$
2. there exists no H_x such that
   (a) $H_x \rightarrow H_i$
   (b) and $H_j \rightarrow H_x$.

Proof of Theorem A:

1. Let $H_{i,M} < N$
   (a) Then $H_{j,M} = H_{i,M+1}$
   and $H_{j,p} = H_{i,p}$
   where $M > p \geq 1.$
   (b) It is seen that

$$/H_j/ - /H_i/ = 1$$

and the theorem is true for the case where

$$H_{i,M} < N.$$

2. Now let $H_{i,M} = N.$
   (a) Then $H_{i,M-k-1} \leq N\text{-}k$
   where $M - k \geq 1.$

This is true from Equation (6).

(b) Starting at

$$k = 1$$

increment k by one until the following occurs:

$$H_{i,M-k-1} < N-k. \qquad (9)$$

This may always be done because, if necessary, let

$$k = M + 1$$

then

$$H_{i,1} < N-M-1$$

which is true from the theorem statement.

(c) At that value of k for which Equation (9) is first true, stop incrementing k.

(d) Let $H_{j,p} = H_{i,p}$

for $K = i;\ p > 1.$

(e) Let $H_{j,r} = H_{j,r+1} + 1$

(f) where $1 < r \geq k.$

(g) From Equations (6) and (8), it is true, by construction, that no

$$H_x \neq H_j$$

can be constructed such that

(1) $H_x \rightarrow H_i$

(2) and $H_j \rightarrow H_x$

because

$$/H_j/ - /H_i/$$

is as small as possible by construction. Corollary A-1. If

$$H_1 = <1, 2, 3, \ldots, (M-1), M>$$

and

$$H_z = <(N-M+1), \ldots, (N-1), N>$$

then by successive application of Theorem A, S_2 through S_z may be found such that

$$H_z \rightarrow H_{z-1} \rightarrow \ldots \rightarrow H_2 \rightarrow H_1$$

and there exists no H_y such that

$$H_{i+1} \rightarrow H_y \rightarrow H_i.$$

Theorem B. The set of H_z through H_1 described in Corollary A-1 is an exhaustive enumeration of the possible combinations of M elements taken at a time from the set S.

Proof:

Realize that the process of forming the H_i may be viewed as first forming a combination and then ordered its elements according to Equation (6). There is no restriction placed on forming the combination but rather on its ordering. Starting with H_1 there is no combination, H_y, which satisfies

$$H_1 \rightarrow H_y.$$

This may be seen to be true using Equations (6) and (8). Similarly, there is no S_y such that

$$H_y \rightarrow H_z,$$

and by Theorem A there is only one combination which may be formed such that

$$H_2 \rightarrow H_1.$$

In general, there is only one S_{i+1} such that

$$H_{i+1} \rightarrow H_i$$

where $z > i \geq 1.$

Thus, all combinations may be formed and then ordered as in steps (1) and (2) of Theorem A which satisfy Corollary A-1 with none being omitted. This completes the proof of the combination algorithm.

Cost and Effectiveness Calculation

The expected effectiveness, in pounds, of an attack may be calculated by multiplying the current state of the target times the percent damage expected on the target, remembering that state is measured as the pounds of supplies interdicted by complete destruction of the target.

The expected cost, in dollars, is equal to the ordnance, fuel, and routine maintenance costs added to the cost of the aircraft and of the pilot* multiplied by the percent damage expected on the attacker.

* According to information obtained from Eglin, Air Force accounting procedures listed the cost of training a pilot as $100,000.00. This is used as the cost of a pilot's life unless a higher value is desired in light of utility theory.

If <u>different targets</u> are to be attacked, the total effectiveness is the sum of the effectivenesses of the individual attacks; the same is true for costs.

All of the above follow immediately from certain algebra of expectations theorems. Notably, the expected value of an event is equal to the event's value multiplied by its probability of occurrence. Further, the attacks on different targets are assumed to be independent events; thus, the expected values are additive.

It is now necessary to examine the case in which the same target is being attacked by more than one weapon system. The only other case is the one presented above in which different targets were being attacked. Assume M planes are attacking. Establish the following notation:

$\underline{S_i}$ is the state of the target prior to the i^{th} attack.

$\underline{A_{i,j}}$ is the percent damage expected to be inflicted during the i^{th} attack by weapon system type "j".

$\underline{B_{i,j}}$ is the percent damage expected to be inflicted on weapon system type "j" during the i^{th} attack.

$\underline{E}$ is the total effectiveness of all the attacks on the target.

$\underline{C_j}$ is the cost of the plane type (e.g., B-58) and pilot used for weapon system "j".

$\underline{O_j}$ is the cost of fuel, routine maintenance, and ordnance of weapon system type "j".

$\underline{C}$ is the total cost for all attacks on the target.

The two basic assumptions of this algorithm are given below:

<u>Assume</u> that the $A_{i,j}$ does not change after the first attack.

$$A_{1,j} = A_{i,j}$$

where

$$i > 1.$$

<u>Assume</u> that the target's defensive ability changes in the following manner:

$$B_{i,j} = \left(\frac{S_i}{S_1}\right) B_{1,j} \qquad (10)$$

for all "j" and all "i".

These assumptions indicate that the ability of the weapon systems to attack a target does not increase after a target has been attacked once, but that the ability of the enemy to defend the target decreases after the first attack. It is assumed that a target remains as difficult to destroy after the first attack as before. This assumption is made to take into account such things as terrain factors and weather conditions. In assuming that the enemy's defensive capability decreases, it is assumed that the percent of target defenses destroyed

is equal to the percent of supplies destroyed. The assumptions are but a first order modeling of reality.

Let A_i be the percent damage expected on the target from the i^{th} attacking weapon system.

Since A_i is the expected percent of the target destroyed, then $(1-A_i)$ is the amount of the target surviving. Thus, the following is true:

$$S_i = S_1 \prod_{j=1}^{i-1} (1-A_j). \tag{11}$$

If M attacks were to be made, then

$$E = S_1 - S_M$$

or

$$E = S_1 - S_1 \prod_{j=1}^{M} (1-A_j). \tag{12}$$

The total cost of an attack is dependent on the attack order. This is because the defensive ability of the target is assumed to decrease as the number of attacks increase. The expression for C may be written

$$C = \sum_{i=1}^{M} \left(\frac{S_i}{S_1}\right) (B_{1,j})\, C_j + 0_j \tag{13}$$

where the subscript "j" assumes its appropriate value for the i^{th} attack.

The optimal ordering of weapon systems attacking the same target will now be considered.

<u>Optimal Ordering of an Attack Permutation</u>

There appears, in general, to be no method for ordering an attack combination into an optimal attack permutation using the C-E criterion other than trial and error. However, if different targets are being attacked, there need be no reordering as the expectation of a sum is equal to the sum of the expectations for independent events.

The following example will illustrate an instance in which an optimal permutation cannot be found by other than trial-and-error methods.

Consider an attack made on one target by three weapon systems: "1", "2", and "3". Let C_i be the cost of the aircraft type and pilot used by the i^{th} weapon system; let A_i be the probability of kill of the target by the i^{th} weapon; and let B_i be the probability of kill of the i^{th} weapon system by the target if that weapon system were to attack first. Neglect ordnance, fuel, and routine maintenance cost as they are always expended and do not affect optimal ordering. This may be seen from Equation (14). Let the initial state of the target be 100 pounds. Let the following data represent the system.

WEAPON SYSTEM	C_i	A_i	E_i
1	10	0.5	0.4
2	9	0.6	0.3
3	8	0.7	0.6

Using the Equation (12), it is found that regardless of attack order, the total effectiveness is 94. The costs for each of the following ordered attacks are:

ATTACK PERMUTATION	COST
$\langle 1, 2, 3\rangle$	6.310
$\langle 1, 3, 2\rangle$	6.805
$\langle 2, 1, 3\rangle$	5.260
$\langle 2, 3, 1\rangle$	5.100
$\langle 3, 1, 2\rangle$	6.405
$\langle 3, 2, 1\rangle$	6.090

Thus, $\langle 2,3,1\rangle$ is the optimal attack order found by forming all permutations. In the event of a tie, only the first least cost ordering is stored by the computer to save space.

In cases wherein aircraft can be ordered in ascending cost and their respective probabilities of being killed when attacking in that order are monotonically non-increasing, then the solution would be to send in the aircraft in that least cost ordering.

For example, for some target and some set of weapon systems, let

$$C_1 \le C_2 \le C_3 \le \ldots. \le C_n.$$

For the attack order $\langle 1,2,3,\ldots,N\rangle$, assume that the probabilities of the weapon systems being killed when attacking in that order may be represented by

$$P_1 \ge P_2 \ge P_3 \ge \ldots. \ge P_N$$

where P_j represents the probability of the j^{th} attacking weapon system being killed.

The following is true:

$$C_1 - C_2 \le 0$$

$$P_2 - P_1 \le 0$$

$$(C_1 - C_2)(P_2 - P_1) \ge 0$$

$$C_1P_2 + C_2P_1 \ge C_1P_1 + C_2P_2 \qquad (14)$$

From Equation (14), it costs less to send in the lower cost weapon system first. Using pair-wise comparisons, it may be seen

that the least cost attack order is

$$<1, 2, 3, \ldots, N>. \qquad (15)$$

Unfortunately, as the prior numerical example has shown, the conditions used in obtaining the above ordering may not always be true. For lower cost aircraft may be expected to have less anti-defection devices and less maneuverability and speed for evasion purposes.

In the computer model, the following is done in cases where more than one aircraft attacks the same target:

1. Arrange the aircraft in order of ascending cost.
2. Calculate the P_j for this ordering.
3. If the P_j are monotonically non-decreasing, then the above ordering is optimal.
4. If not, then all permutations are generated.

Deletion of a Combination

It will be shown how certain attacks can be immediately deleted from consideration using the cost-effectiveness criterion. This results in a limited search rather than an exhaustive search by reducing the initial set of attacks not known to be infeasible. The limited search aspect is a major advantage of the proposed system.

Establish the following terminology:

i, j, k, r are arbitrary positive integers.

$A_i^{j,r}$ is the probability of target "r" being killed by weapon system "i" after the target has already been attacked "j-i" times by weapon system "i".

$B_i^{j,r}$ is the corresponding probability of the weapon system being killed by the target.

$T_i^{j,r}$ is the total cost incurred in the above attack.

$E_i^{j,r}$ is the effectiveness obtained by the above attack.

O_i is the cost of the aircraft and pilot neglecting ordinance, fuel, and routine maintenance.

N_i is the availability of weapon system "i".

M is the total number of aircraft to be sent.

$D_{i,r}$ is the number of type "i" weapon systems known to be infeasible for an attack on target "r".

The defense capability of target "r" is assumed to be linearly

proportional to the percent damage already inflicted on it. By this assumption it may be inferred that

$$A^2_{m,r} = (A^1_{m,r})y$$

where y is a positive constant of proportionality. It follows that

$$A^j_{m,r} = (A^1_{m,r})y^{j-1}.$$

Let $A^1_{m,r} > A^1_{n,r}$,

then

$$A^j_{m,r} > A^j_{n,r}.$$

It is recognized that the assumption used may be but a first order approximation to reality. Its use enables a significant saving in computation time by reducing the feasible set of solutions to be considered with regard to the cost effectiveness criterion.

Effectiveness is equal to the state of the target multiplied by the probability of its being killed. Thus, by the same operations as above,

$$E^j_{m,r} > E^j_{n,r},$$

weapon system "m" causes a higher percent damage in every attack than does weapon system "n".

It is seen that letting weapon system "m" attack target "r" at any time is preferable from effectiveness considerations to the target being attacked by weapon system "n", because it will increase the total effectiveness of an attack permutation.

Now let cost considerations be examined. Let the following be true:

$$0_m < 0_n$$

and

$$T^1_{m,r} < T^1_{n,r}$$

then the following is true

$$0_m + B^1_{m,r}C_m < 0_n + B^1_{n,r}C_n$$

and

$$0_m - 0_n < B^1_{n,r}C_n - B^1_{m,r}C_m.$$

Let "x" be some number less than or equal to one, and let it represent the percent of target "r" surviving after "j-1" attacks. Then assume that the following:

$$B^j_{i,r} = (1-x)B^1_{i,r}.$$

This defines the defensive ability of a target as equal to its original defensive ability times the amount of destruction the target

has absorbed.

It is true that

$$(0_m - 0_n) < (x)\,(B^1_{n,r}C_n - B^1_{m,r}C_m)$$

because the left member of the inequality is less than zero and the right member, if negative, is increased and, if positive, is still positive. Rearranging terms:

$$0_m + (x)\,(B^1_{m,r})C_m < 0_n + (x)(B^1_{n,r})C_n$$

or $$T^j_{m,r} < T^j_{n,r}\ .$$

From cost consideration, $A^j_{m,r}$ is preferable to $A^j_{n,r}$ as it will reduce the cost of an attack permutation.

Using the cost-effectiveness criterion, an attack on target "r" by weapon system "m" is preferable to an attack on target "r" by weapon system "n" since it has lower cost but higher effectiveness if the following are true:

$$A^j_{m,r} > A^j_{n,r} \tag{16}$$

$$0_m < 0_n \tag{17}$$

$$T^1_{m,r} < T^1_{n,r}. \tag{18}$$

The immediate deletion algorithm shown in Figure 3 finds attacks preferable from a cost-effectiveness viewpoint.

The number of attacks that can be deleted from consideration is shown by the following example.

For target "r", let there be types of weapon systems i, j, and k with availabilities 3, 3, and 1, respectively. Assume five planes are scheduled to be sent. For target "r", let weapon systems "i" be superior to both "j" and "k" with respect to the cost effectiveness criterion, and let "j" be superior to "k". Then it may be seen that:

$$D_{j,r} = N_j-(M-N_i) \tag{19}$$

if $$N_j \geq M-N_i$$

and $$D_{j,r} = 0 \tag{20}$$

if $$N_j \leq M-N_i.$$

Similarly, it may be seen that

$$D_{k,r} = N_k-(M-N_i-N_j) \tag{21}$$

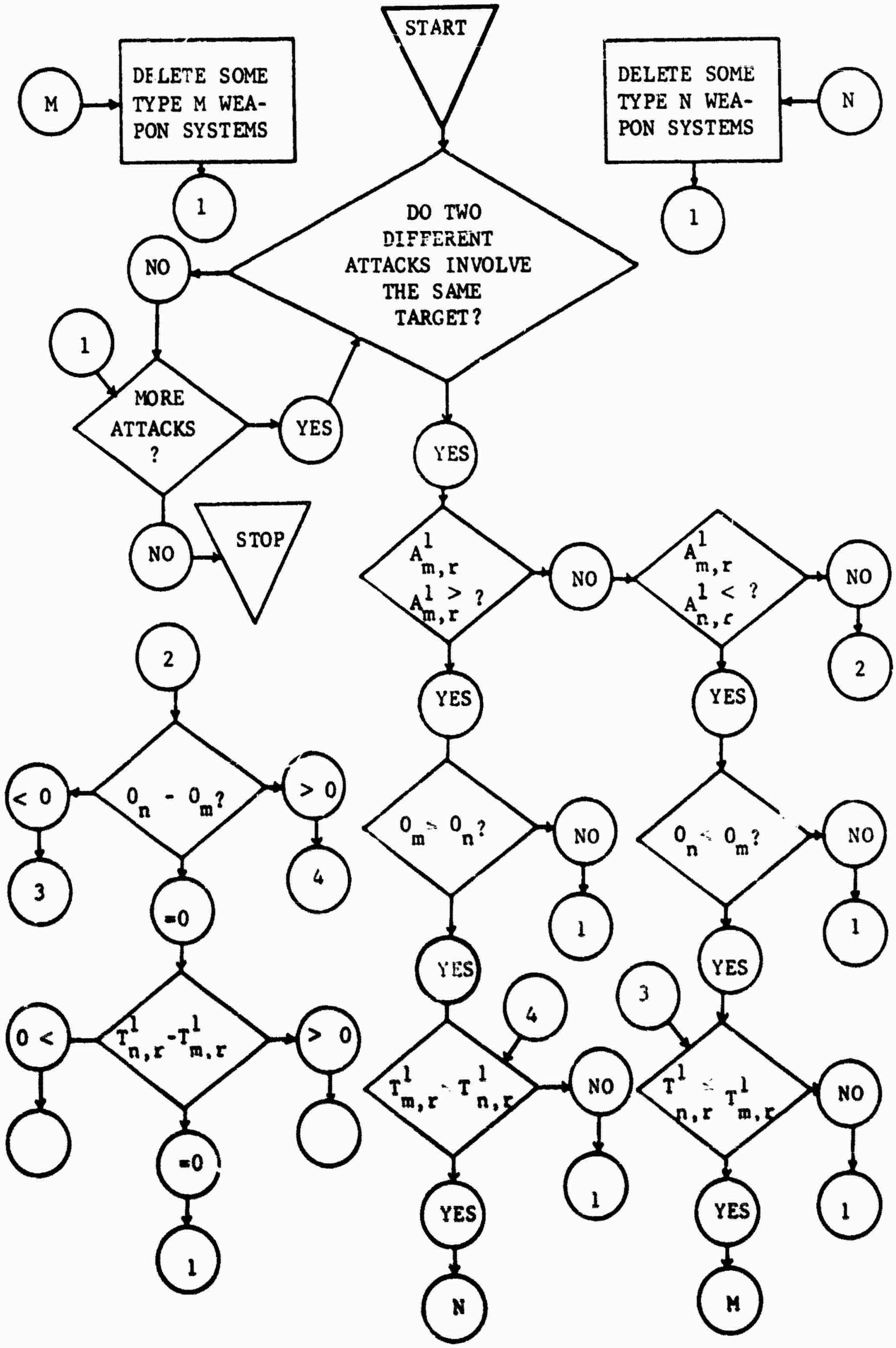


Figure 3. Immediate Deletion Algorithm

if $\quad N_k \geq M-N_i-N_j$

and $\quad \text{[illegible]}_{k,\text{[illegible]}} = 0 \qquad (22)$

if $\quad D_k \leq M-N_i-N_j.$

The above analysis is restricted to weapon systems attacking the same target. If different targets were to be attacked, it is not possible here to predict, from initial conditions, what the defensive and vulnerability characteristics of two different targets will be if more than one plane is to be sent. For after the first attacks, the characteristics of the attacked targets change relative to one another if the same target is not being attacked.

SECTION V

EXAMPLE OF COMPUTER ALGORITHM

A complete example of the computer algorithm will be presented here. This will include assuming the tactical situation, input data, and the solution.

Consider that the network shown in Figure 4 is to be interdicted:

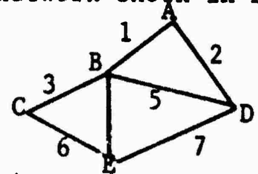


Figure 4. Hypothetical Supply Network

Let there be two different types of aircraft available to attack the network: aircraft type 1 and aircraft type 2. Further, assume that there are two aircraft of type 1 and two of type 2 available. Let it be possible that each type of aircraft may be armed in two different ways. For the purposes of the algorithm, there are six weapon systems subject to aircraft availabilities which may be used. There are three different types of weapon systems. This may be seen in Table I. Call the two weapon systems which use aircraft type 1, weapon systems 11 and 12, weapon system 18.

Assume that the commander, rather than wishing to examine all possible attacks against the network, chooses to attack the storage depot on branch 1 or to attack either branch 2 itself or the vehicles thereon. Assume that all necessary data is known and that the commander wishes to send three aircraft. The algorithm will, in this case, show the cost and effectiveness of all possible permutations of three attacks which are not deleted using the cost-effectiveness criterion. The data is presented in Tables 1, II, and IV.

Tables I and II give the weapon system data and the network data necessary for this problem. For purposes of this example, ordnance, fuel, and routine maintenance costs are assumed not to differ for the various targets. This is not always the case.

TABLE I. WEAPON SYSTEM DATA

W/S TYPE	AVAILABILITY	PLANE TYPE	PLANE TYPE COST (DOLLARS)	ORDNANCE, FUEL, AND ROUTINE MAINTENANCE COST (DOLLARS)
11	2	1	800,000.00	90,000.00
12	2	1	800,000.00	100,000.00
18	2	2	900,000.00	120,000.00

TABLE I. WEAPON SYSTEM DATA (Concl'd)

W/S TYPE	AVAILABILITY	PLANE TYPE	PLANE TYPE COST (DOLLARS)	ORDNANCE, FUEL, AND ROUTINE MAIN-TENANCE COST (DOLLARS)
PLANE TYPE	AVAILABILITY			
1	2			
2	2			

TABLE II. NETWORK DATA

BRANCH	VEHICLES EXPECTED	AVERAGE VEHICLE CAPACITY (POUNDS)	AVERAGE VEHICLE FLOW RATE (VEH./HR.)	TIME TO REBUILD (HOURS)	SUPPLIES STORED (POUNDS)
1	2	100	10	30	50
2	3	70	8	20	70

The state of a storage depot is equal to the pounds of supplies stored there multiplied by the relative weighting factor assigned to those supplies.

The state of the vehicles (when considered as targets) is equal to the number of vehicles expected to be on the route at the time of attack multiplied by the average vehicle capacity (in pounds), multiplied by the relative weighting factor assigned to those supplies.

The state of a branch (when considered as a target) is equal to the product of average vehicle capacity (in pounds), the average vehicle flow rate (in vehicles per hour), and the time to rebuild the branch (in hours) if it were completely destroyed. This product is multiplied by the relative weighting factor to obtain the final value for the state of the branch.

Using the above relationships, the various states are calculated and are presented in Table III. For the example problem, the targets are assumed to contain supplies of equal combat utility to the enemy and are all assigned a weighting factor of 1.0.

TABLE III. TARGET STATES

BRANCH	TARGET	STATE
1	SUPPLIES	50.
1	VEHICLES	200.
1	BRANCH	30,000.
2	SUPPLIES	70.
2	VEHICLES	210.
2	BRANCH	12,000.

In Table IV the attack data is given. This data is given only for those attacks which are under consideration. The attack number shown in Table IV is not data which would be read into the computer. It is presented here for illustrative purposes and is a number internally generated by the computer. The number to the right of the targets is used to identify the target by a number, which is used only for the purpose of this example.

TABLE IV. ATTACK DATA

ATTACK NUMBER	W/S	BRANCH	TARGET	PROBABILITY % OF BEING KILLED	PROBABILITY OF KILLING	WEIGHT
1	11	1	SUPPLIES	.40	.30	1.0
2	12	1	SUPPLIES(1)	.35	.15	1.0
3	18	1	SUPPLIES(1)	.30	.18	1.0
4	11	2	BRANCH(2)	.20	.05	1.0
5	12	2	BRANCH(2)	.15	.10	1.0
6	18	2	BRANCH(2)	.12	.15	1.0
7	11	2	VEHICLES(3)	.15	.20	1.0
8	12	2	VEHICLES(3)	.20	.15	1.0
9	18	2	VEHICLES(3)	.18	.10	1.0

The information in the Tables I, II, and IV (weapon system, network, and attack data) is the complete set of data (excluding the attack numbers) which is needed for the computer model. In Appendix I the system by which this data is put onto data cards for input into a computer is described.

Here all weights are equal to 1.0. If any weights were not equal to this value, the computer model would internally adjust the state of a target with a weighting factor differing from unity to its appropriate value. This weighting factor may be determined subjectively or by some other methods. For example, assume that a storage location contains 100 pounds of supplies with a weighting factor of 2.0. The model would compute the state of the target to be 200.

The first step of the algorithm is to determine if it is possible to delete immediately any attacks from consideration. The flowchart for this process is shown in Figure 3. In order to do this, the cost of each attack must be calculated as though it were the first attack. The cost of a first attack would be the ordnance, fuel, and routine maintenance cost of the weapon system added to the cost for aircraft and for pilot multiplied by the probability of their being killed. This may be seen from Equation (13). Table V gives these costs for the example under consideration.

TABLE V. MISSION COSTS

ATTACK NUMBER	MISSION COST (DOLLARS)
1	410,000.00
2	380,000.00
3	390,000.00
4	250,000.00
5	220,000.00
6	228,000.00
7	210,000.00
8	260,000.00
9	282,000.00

The first three attacks involve the same target, the second three another, and the third three another. Thus, there are three groups of comparisons which must be made to ascertain if any attacks may be immediately deleted from consideration. Referring to Equations (16), (17), and (18), a deletion is possible if any of the following are true:

$$A^{1}_{m,r} > A^{1}_{n,r}$$

$$O_{m} < O_{n}$$

$$T^{1}_{m,r} < T^{1}_{n,r}.$$

The attack indicated on the right-hand side of the inequalities would be deleted.

For the first set of comparisons:

$$A^{1}_{11,1} > A^{1}_{12,1} > A^{1}_{18,1}$$

$$O_{11} < O_{12} < O_{18} \tag{23}$$

$$T^{1}_{12,1} < T^{1}_{18,1} < T^{1}_{11,1}$$

Using Equation (20) it is seen that:

$$D_{18,1} = N_{18}-(M-N_{11})$$

$$D_{18,1} = 2-(3-2)$$

$$D_{18,1} = 1.$$

From Table I, it is known that there were originally two weapon systems capable of performing attack number 3. From the above analysis, it is seen that one may be deleted from consideration by using the immediate deletion algorithm. This is the only deletion which may be performed using the first set of comparisons.

The following is true for the second set of comparisons:

$$A^1_{12,2} > A^1_{18,2} > A^1_{11,2}$$

$$O_{11} < O_{12} < O_{18}$$

$$T^1_{12,2} < T^1_{18,2} < T^1_{11,2}.$$

There are no deletions that may be performed.

The following is true for the third set of comparisons:

$$A^1_{11,3} > A^1_{12,3} > A^1_{18,3}$$

$$O_{11} < O_{12} < O_{18}$$

$$T^1_{11,3} < T^1_{12,3} < T^1_{18,3}.$$

Originally two aircraft were capable of performing attack number 8. It is shown below that one of these may be deleted from consideration by using Equation (19). Similarly, two aircraft were originally capable of performing attack number 7. It is shown that both of these may be deleted from consideration by using Equation (22). For attack number 8:

$$D_{12,3} = N_{12}-(M-N_{18})$$

$$D_{12,3} = 2-(3-2)$$

$$D_{12,3} = 1.$$

For attack number 7:

Since N_{11} $M-N_{12}-N_{18}$

or $2 \geq -1$

then $$D_{11,3} = N_{11}$$

$$D_{11,3} = 2.$$

Now the complete list of attacks not known to be infeasible may be given in Table VI.

TABLE VI. ATTACK NUMBERS AFTER DELETION

ATTACK NUMBER	BRANCH	TARGET	W/S TYPE
1	1	SUPPLIES (1)	11
2	1	SUPPLIES (1)	11
3	1	SUPPLIES (1)	12
4	1	SUPPLIES (1)	12
5	1	SUPPLIES (1)	18
6	2	BRANCH (2)	11
7	2	BRANCH (2)	11
8	2	BRANCH (2)	12
9	2	BRANCH (2)	12
10	2	BRANCH (2)	18
11	2	BRANCH (2)	18
12	2	VEHICLES (3)	12
13	2	VEHICLES (3)	18
14	2	VEHICLES (3)	18

Temporarily neglect availability constraints. From the above set of fourteen attack numbers, all combinations of three elements taken at a time must be obtained. For each of these combinations, the optimal attack permutation must be obtained:

Referring to Equation (2), it is seen that there are 364 combinations, since three elements at a time are being taken from the reduced set of 14 attacks.

$$C_3^{14} = \frac{14!}{3!(14-3)!}$$

$$C_3^{14} = 364$$

For each of these combinations which involve attacks on the same target, permutations must be considered.

Before the preliminary deletions presented above, there were 544 combinations since there were originally 18 possible attacks before deletion.

$$C_3^{18} = \frac{18!}{3!(18-3)!}$$

$$C_3^{18} = 544$$

For each of the 544 combinations which involve attacks on the same target, permutations must also be considered. Because the preliminary deletion eliminated only attacks on the same target, then there are less permutations to be considered after deletion than before. As an upper bound on the reduction of total number of attack permutations to be investigated as a result of preliminary deletion which reduced the original 544 combinations to 364, the following is true:

$$\text{REDUCTION} = \frac{544-364}{544}$$

$$\text{REDUCTION} = 35\%.$$

This is a significant reduction in the work necessary to investigate completely this particular problem.

Using the attack numbers given in Table VI, the following attack combinations may be developed:

TABLE VII. ATTACK COMBINATIONS

COMBINATION NUMBER	COMBINATION
1	1, 2, 3
2	1, 2, 4
.	.
.	.
.	.
12	1, 2, 14
13	1, 3, 4
14	1, 3, 5
.	.
.	.
.	.
23	1, 3, 14
24	1, 4, 5
.	.
.	.
.	.
34	1, 5, 6
.	.
.	.
.	.
40	1, 5, 12
.	.
.	.
.	.
48	1, 6, 12
.	.
.	.
.	.

TABLE VII. ATTACK COMBINATIONS (Concl'd)

COMBINATION NUMBER	COMBINATION
78	1, 13, 14
79	2, 3, 4
80	2, 3, 5
.	.
.	.
.	.
362	11, 12, 13
363	11, 12, 14
364	12, 13, 14

Now consider the availability constraints. Examine combination numbers 1 and 2. Both of these require that three aircraft of type 1 be used. Both of these combinations would be infeasible with regard to aircraft availability considerations. Before the cost and effectiveness values for any combination are evaluated, the feasibility of that combination, with regard to aircraft type and to weapon system type availabilities, is checked. If a combination violates availability constraints, it is deleted from consideration.

Further, it is seen that combinations number 1 and 2 constitute an identical attack combination, i.e., the same weapon systems attack the same target. Due to the manner in which the information storage and retrieval subsystem of the computer model is constructed, one or more identical combinations immediately follow the first. The computer model is designed so that, immediately after investigating the first of an identical set of combinations, it skips to a new combination and examines it in order to avoid duplication of work.

To complete this example problem, only three attack combinations will be examined, evaluated, and compared. This will be sufficient to illustrate the method. The actual computer model would examine all combinations. The computer solution to this problem is given in Appendix II. The input data has been changed such that the program examines the possibility of sending one through four planes (inclusive) instead of only examining the possibility of sending three. Appendix I shows how to introduce the input data onto computer cards for the problem of Appendix II.

The following attack numbers are investigated below: combination numbers 34, 40, and 48.

For attack combination number 34, two permutations must be considered because attack numbers 1 and 5 involve the same target. Referring to Table VI, it is seen that attack numbers 1 and 5 both

involve attacks on the supplies of branch number 1. Attack number 6 involves an attack on branch number 2. Equation (13) is used for effectiveness calculations, and Equation (14) is used for cost calculations when the attacks involve the same target.

The effectiveness of attack number 6 is equal to the probability of killing the target multiplied by the state of the target. This value is 600.0. The cost is equal to the mission cost as calculated for Table V. Attack number 6, as read from Table VI, corresponds to attack number 4 as read from Table V. (Table V lists only unique attacks whereas Table VI lists all feasible attacks.) This value is 250,000.0.

Now consider the effectiveness that would result from letting attack number 1(as read from Table VI) occur first, followed by attack number 5. The effectiveness and cost values for attack number 1 would be found in the same manner as those of attack number 6. These values are 15.0 and 410,000.0, respectively. The attack performed in attack number 1 would lower both the state of the target and its defensive ability by 30%. Thus, immediately before attack number 5 is performed, the state of the target is 35.0. The probability of weapon system 18 being killed in the attack is 70% of the original probability or 0.21. Using these values of state and defensive ability, the effectiveness and cost values obtained from attack number 5 are 6.3 and 302,000.0, respectively. These values are relative and must be seen in combination with other choices.

The total effectiveness and cost values for this permutation are obtained by summing the above values for each attack. Total effectiveness is 621.3, and total cost is 962,000.0 for this permutation of attack combination number 34.

The total effectiveness and total cost values for the other permutation of attack combination number 34 are 621.3 and 992,400.0, respectively. Thus, using the cost effectiveness criterion, the first combination is preferable to the second and is the only one stored in the computer's memory.

By proceeding in a similar manner, all other attack permutations may be evaluated. The results are presented in Table VIII.

TABLE VIII. COST AND EFFECTIVENESS VALUES

ATTACK PERMUTATION	EFFECTIVENESS	COST
<1, 5, 6>	621.3	962,000.0
<5, 1, 6>	621.3	992,400.0
<1, 5, 12>	52.8	972,000.0
<5, 1, 12>	52.8	1002,400.0
<1, 6, 12>	646.5	920,000.0

Using the cost-effectiveness criterion with an effectiveness tolerance of 0.0 pound and a cost tolerance of 0.0 dollars, it may be seen from Table VIII that the attack permutation <1, 6, 12> is the only attack combination of the original three which was considered. The attacks of the attack combination <1, 6, 12> may be permuted in any order because three separate targets are attacked, and the cost and effectiveness calculations for each one are independent.

In this example a tolerance level of 0.0 dollars and 0.0 pound is used. This, in fact, indicates absolute certainty regarding the probabilities of kill used to find expected values and regarding other data. If uncertainty is present, it is advisable to increase the tolerance levels.

SECTION VI

CONCLUSIONS AND RECOMMENDATIONS

The computer model is capable of converting a large amount of data to assist a field commander in his decision processes. The results of potential attack combinations and attack permutations are evaluated and compared so that interdiction can be accomplished efficiently.

Up-to-date and accurate data are necessary inputs into the model in order to have meaningful results. After a series of attacks has been completed, the data for any affected part of the supply network must be updated. This is due to the short run nature of the algorithm.

It is recommended that the long run problem in which network characteristics may vary over time be investigated. By means of such an investigation, information relevant to strategic planning would be made available.

APPENDIX I

COMPUTER PROGRAM DOCUMENTATION

Below is a computer listing explaining preparation of input data:

```
CARD  COLUMN  DATA (AND FORMAT)  DECIMAL MUST BE SUPPLIED FOR 'F' FORMATS
                                 'I' FORMATS MUST BE RIGHT JUSTIFIED
  1     1-5   THE MINIMUM NUMBER OF PLANES TO BE SENT (I5)
  1     6-10  THE MAXIMUM NUMBER OF PLANES TO BE SENT (I5)
  2       5   PUT '1' TO OUTPUT  IN ORDER OF DECREASING EFFECTIVENESS
              PUT '2' TO OUTPUT IN ORDER OF INCREASING COST
              PUT '3' TO HAVE BOTH OF THE ABOVE DONE
  2     6-10  THE MAXIMUM NUMBER OF ALTERNATIVES TO BE OUTPUT FOR ANY NUMBER
                  OF PLANES TO BE SENT. LEAVE BLANK IF ALL ARE TO BE PRINTED
                  (I5)
  2    11-20  THE VALUE BY WHICH ANY TWO EFFECTIVENESS VALUES MAY DIFFER BEFORE
                  THEY ARE TO BE CONSIDERED DIFFERENT (F10.0)
  2    21-31 THE VALUE BY WHICH ANY TWO COST VALUES MAY DIFFER BEFORE THEY ARE
                  CONSIDERED TO BE DIFFERENT (F10.0)
  2      35   PLACE '1' IF DELETION IS DESIRED, LEAVE BLANK OTHERWISE (I5)
----  ------  REPEAT THE FOLLOWING SEQUENCE AS LONG AS THE DESCRIPTION IS
                  APPLICABLE. EACH CARD SHOULD HAVE FOUR DATA ELEMENTS ON IT
        1-5   WEAPON SYSTEM TYPE NUMBER (I5)
        6-10  THE AVAILABILITY OF THE ABOVE WEAPON SYSTEM (I5)
       11-15  THE PLANE TYPE NUMBER OF THE PLANE USED FOR THE ABOVE W/S (I5)
       16-25  THE DOLLARS COST OF THE ABOVE WEAPON SYSTEM (F10.0)
----  ------  REPEAT THE FOLLOWING SEQUENCE AS LONG AS THE DESCRIPTION IS
                  APPLICABLE. EACH CARD SHOULD HAVE TWO DATA ELEMENTS ON IT
        1-5   PLANE TYPE NUMBER (I5)
        6-10  THE AVAILABILITY OF THE ABOVE PLANE TYPE (I5)
----  ------  REPEAT THE FOLLOWING SEQUENCE AS LONG AS THE DESCRIPTION IS
                  APPLICABLE. EACH CARD SHOULD HAVE SEVEN DATA ELEMENTS ON IT
        1-10  BRANCH NUMBER (I10)
       11-20  THE NUMBER OF VEHICLES EXPECTED ON THE ABOVE BRANCH (F10.0)
       21-30  THE AVERAGE POUNDS OF SUPPLIES CARRIED BY EACH VEHICLE ON THE
                  ABOVE BRANCH (F10.0)
       31-40  THE AVERAGE NUMBER OF VEHICLES PER HOUR WHICH FLOW ALONG THE
```

```
                ABOVE ROUTE (F10.0)
   41-50   THE AVERAGE MPH OF THE VEHICLES ON THE ABOVE BRANCH (F10.0)
   51-60   THE HOURS TO REBUILD THE ABOVE BRANCH IF COMPLETELY DESTROYED
                (F10.0)
   61-70   THE POUNDS OF SUPPLIES STORED ON THE ABOVE BRANCH (F10.0)
---- ------ REPEAT THE FOLLOWING SEQUENCE AS LONG AS THE DESCRIPTION IS
                APPLICABLE. EACH CARD SHOULD HAVE SEVEN DATA ELEMENTS ON IT
    1-5    WEAPON SYSTEM TYPE NUMBER WHICH IS TO PERFORM AN ATTACK (I5)
    6-10   THE BRANCH WHICH THE ABOVE W/S IS TO ATTACK (I5)
     15    PLACE A '1' IF A SUPPLY DEPOT IS TO BE ATTACKED BY THE ABOVE W/S
           PLACE A '2' IF VEHICLES ARE TO BE ATTACKED
           PLACE A '3' IF A BRANCH IS TO BE ATTACKED
   16-25   THE PROBABILITY THAT THE ABOVE W/S WILL BE DESTROYED BY THE ABOVE
                TARGET (F10.0)
   26-35   THE PROBABILITY THAT THE ABOVE TARGET WILL BE DESTROYED BY THE
                ABOVE WEAPON SYSTEM (F10.0)
   36-45   THE TOTAL DOLLARS COST FOR FUEL, ROUTINE MAINTENANCE, AND
                ORDNANCE FOR THE ABOVE ATTACK (F10.0)
   46-55   THE RELATIVE WEIGHTING FACTOR FOR THE TARGET OF THE ABOVE ATTACK.
                IF ALL SUPPLIES AT ALL TARGETS ARE TO BE CONSIDERED EQUALLY
                IMPORTANT, 1 MAY BE PLACED HERE (F10.0)
```

```
                  BELOW IS THE INPUT FOR THE EXAMPLE COMPUTER PROGRAM
                  THE FIRST TWO PRINTED LINES INDICATE COLUMN NUMBER
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

    1    4
    1    0   0.0       0.0       1
    3    2    2
   11    2    1  800000.
   12    2    1  600000.
   18    2    2  900000.
    1    2
    2    2
         1    2.     100.      10.       20.       30.       50.
         2    3.      70.       8.       15.       20.       70.
   11    1    1      .30       .40     90000.      1.0
   12    1    1      .15       .35    100000.      1.0
   18    1    1      .18       .30    120000.      1.0
   11    2    3      .05       .20     90000.      1.0
   12    2    3      .10       .15    100000.      1.0
   18    2    3      .15       .12    120000.      1.0
   11    2    2      .20       .15     90000.      1.0
   12    2    2      .15       .20    100000.      1.0
   18    2    2      .10       .18    120000.      1.0
-1
```

BELOW IS A COMPUTER LISTING OF AN EXPLANATION OF VARIABLE NAMES

```
AVCAP  TEMPORARY STORAGE OF THE AVERAGE VEHICLE CAPACITY ON A BRANCH
AVER  TEMPORARY STORAGE OF THE AVERAGE NUMBER OF VEHICLES PER HOUR ON A BRANCH
AVSPD  TEMPORARY STORAGE IN MPH OF THE AVERAGE SPEED OF THE VEHICLES ON A BRANCH
C  THE COST OF A SPECIFIED ATTACK COMBINATION
CM  THE COST OF THE ATTACK IN ATTACK NUMBER I4
COLC  IS THE CURRENT LEAST COST VALUE FOR A SET OF PERMUTATIONS
COMBC  SUBROUTINE WHICH GENERATES THE FIRST SET OF ATTACK COMBINATIONS
CORD  THE COST OF ORDNANCE, FUEL, AND MAINTENANCE INCURRED IN ATTACK NUMBER I4
CP  THE COST OF THE PLANE TYPE (INCLUDING PILOT) USED BY A SPECIFIED W/S
CPER( )  IS USED TO STORE THE COST VALUE OF A PERMUATION
CPER1  IS USED TO TEMPORARILY STORE COST VALUES OF A PERMUTATION
CPTEST( )  IS A TEMPORARY STORAGE ARRAY FOR CP( )
CPTST1  IS TEMPORARY STORAGE FOR CPTEST( )
CPX  IS USED TO CALCULATE THE PROBABILITY OF A W/S BEING KILLED
E  THE EFFECTIVENESS OF A SPECIFIED ATTACK COMBINATION
EFFEC  SUBROUTINE WHICH CALCULATES THE EFFECTIVENESS AND COST OF COMBINATIONS
EPER( )  IS USED TO STORE THE EFFECTIVENESS VALUE OF A PERMUTATION
EPER1  IS USED TO TEMPORARILY STORE EFFECTIVENESS VALUES OF A PERMUTATION
HIT  THE POUNDS INTERDICTED BY AN ATTACK IN THE EFFECTIVENESS CALCULATION
HLDMAP  THE ATTACK NUMBERS USED IN THE EFFECTIVENESS CALCULATION AS THEY APPEAR
        IN THE FIRST SUBSCRIPT OF STATE
HOLDST  VALUES OF VARIOUS STATES BEFORE THE EFFECTIVENESS CALCULATION
IC  THE ATTACK COMBINATIONS. THE FIRST SUBSCRIPT DESIGNATES THE ATTACK COMBINA-
    TION AND THE SECOND THE ATTACK NUMBERS FOR THIS COMBINATION
IC1  TEMPORARY STORAGE FOR IC
IC2  A TEMPORARY STORAGE ARRARY OF IC
IG( )  IS AN ARRAY USED TO STORE PERMUTATIONS
IHDOR( )  IS A TEMPORARY STORAGE ARRAY FOR IORDER( ) USED TO STORE THE CURRENTLY
          BEST PERMUTATION
IHDOR1( )  IS A TEMPORARY STORAGE ARRAY FOR IORDER( )
IMDL  IS '0' WHEN IMMDEL HAS NOT BEEN CALLED PREVIOUSLY, '1' OTHERWISE
IMMDEL  IS THE IMMEDIATE DELETION SUBROUTINE
IORDER( )  COTAINS THE ORDERING OF ATTACKS ON A SINGLE TARGET
```

```
IOCER1   IS TEMPORARY STORAGE FOR IORDER( )
IOCER2   IS TEMPORARY STORAGE FOR IORDER( )
ISKIP1   TEMPORARY STORAGE OF ISKIP
ISKIP  THE ATTACK NUMBER TO WHICH THE SUBROUTINE COMBC SKIPS AFTER AN ATTACK
       NUMBER HAS BEEN REJECTED DUE TO AVAILABILITY CONSTRAINTS
ISOL   THE WEAPON SYSTEM USED IN ATTACK NUMBER I4
ITEST  THE ATTACK NUMBERS OF AN ATTACK COMBINATION BEING TESTED FOR FEASIBILITY
ITEST1   TEMPORARY STORAGE FOR ITEST
I1   A SIMPLE COUNTER
I2   A SIMPLE COUNTER
I3   A SIMPLE COUNTER
I4END  THE TOTAL NUMBER OF ATTACKS CONSIDERED
I4   A SIMPLE COUNTER OF THE NUMBER OF ATTACKS READ INTO THE COMPUTER
I5   A SIMPLE COUNTER
I6   A SIMPLE COUNTER
I6END  THE TOTAL NUMBER OF ATTACK NUMBERS FOR SUBROUTINE COMBC
I6END1   IS I6END MINUS '1'
I7   A SIMPLE COUNTER
I8   A SIMPLE COUNTER
I9   A SIMPLE COUNTER
I41   A SIMPLE COUNTER
I45   IS A SIMPLE COUNTER
I46   IS A SIMPLE COUNTER
I451   IS A SIMPLE COUNTER
I51   IS A SIMPLE COUNTER
I55   IS A SIMPLE COUNTER
I56   IS A SIMPLE COUNTER
I61   IS A SIMPLE COUNTER
I71   IS A SIMPLE COUNTER
JSOL   THE BRANCH ATTACKED IN ATTACK NUMBER I4
J1HOLD   THE NUMBER OF ATTACK COMBINATIONS MEETING FEASIBILITY CONSTRAINTS
J1OLD  THE NUMBER OF ATTACK COMBIATIONS FOR NASGN1 PLANES
J1HLD1   EQUALS J1HOLD MINUS 1
J1HLD2   THE NUMBER OF ATTACK COMBINATIONS AFTER DELETION
J1HLD3   EQUALS J1HLD2 MINUS 1
J2HOLD   THE VALUE OF J2 IMMEDIATELY AFTER A FEASIBILITY ACCEPTANCE OR REJECTION
```

```
J1   A SIMPLE COUNTER
J2   A SIMPLE COUNTER
J3   A SIMPLE COUNTER
KASGN  THE MAXIMUM NUMBER OF PLANES TO BE SENT
KATK  MAPS THE EXPANDED SET OF NUMBERS USED IN SUBROUTINE COMBO INTO THE ATTACK
KATK1  TEMPORARY STORAGE FOR KATK
KATK2  IS TEMPORARY STORAGE FOR KATK( )
KAVL( )  IS THE FEASIBLE NUMBER OF W/S USED FOR ATTACK I4
KB  DESIGNATES THE NUMBER OF WEAPON SYSTEMS SUPERIOR TO SOME W/S
KBL  IS TEMPORARY STORAGE FOR KBTR( , )
KBT  IS TEMPORARY STORAGE FOR KBTI( )
KBTI(I)  IS THE NUMBER OF TIMES ANOTHER UNIQUE ATTACK IS SUPERIOR TO ATTACK I
KBTR(I,J)  IS THE ATTACK NUMBER OF THE JTH ATTACK SUPERIOR TO ATTACK I
KHECK  THE MAXIMUM VALUE OF ITEST(J2) SO THAT NOT MORE THAN I6END ATTACK NUMBERS
      NUMBERS READ INTO THE COMPUTER USING THE SUBSCRIPT I4
       ARE CONSIDERED
KPLASN  THE NUMBER OF PLANES ASSIGNED OF A SPECIFIED PLANE TYPE NUMBER
KPLAVL  THE AVAILABILITY OF A SPECIFIED PLANE TYPE NUMBER
KPTYP1  TEMPORARY STORAGE OF PLANE TYPE NUMBERS
KSLHLC  THE TARGETS ATTACKED BY AN ATTACK COMBINATION
KSOL  THE TARGET NUMBER ATTACKED IN ATTACK NUMBER I4
KSOL1  TEMPORARY STORAGE FOR KSOL
KSTOP  THE MAXIMUM NUMBER OF ATTACK COMBINATIONS TO BE OUTPUT EACH TIME
KWSASN  THE NUMBER OF WEAPON SYSTEMS ASSIGNED OF A SPECIFIED TYPE NUMBER
KWSAVL  THE AVAILABILITY OF A SPECIFIED WEAPON TYPE NUMBER
KWSPL  THE PLANE TYPE NUMBER USED FOR A SPECIFIED WEAPON SYSTEM NUMBER
KWSPL1  TEMPORARY STORAGE FOR KWSPL
KWTYPE  W/S TYPE NUMBER OF THE IIST W/S READ INTO THE COMPUTER
KWTYP1  TEMPORARY STORAGE FOR WEAPON TYPE NUMBERS
K1   A SIMPLE COUNTER
K2   A SIMPLE COUNTER
K3   A SIMPLE COUNTER
K4   A SIMPLE COUNTER
K5   A SIMPLE COUNTER
K6   IS A SIMPLE COUNTER
K7   A SIMPLE COUNTER
```

```
KH   A SIMPLE COUNTER
K11   IS A SIMPLE COUNTER
K20   IS A SIMPLE COUNTER
K30   IS A SIMPLE COUNTER
K32   IS A SIMPLE COUNTER
K40   IS A SIMPLE COUNTER
K41   DESIGNATES THE POSITION IN IORDER( ) WHICH CONTAINS THE FIRST OF A SET OF
      ATTACK NUMBERS TO BE PERMUTED
K42   IS A SIMPLE COUNTER
K43   DESIGNATES THE POSITION IN IORDER( ) WHICH CONTAINS THE LAST OF A SET OF
      ATTACK NUMBERS TO BE PERMUTED
K45   IS USED IN FINDING DIFFERENT PERMUTATIONS
K46   IS USED IN FINDING DIFFERENT PERMUTATIONS
K47   IS USED IN FINDING DIFFERENT PERMUTATIONS
K48   IS USED IN FINDING DIFFERENT PERMUTATIONS
K50   IS A SIMPLE COUNTER
K51   IS A SIMPLE COUNTER
K61   IS A SIMPLE COUNTER
K49   IS A SIMPLE COUNTER
K481   IS USED IN FINDING DIFFERENT PERMUTATIONS
K491   IS A SIMPLE COUNTER
K492   IS A SIMPLE COUNTER
K494   IS A SIMPLE COUNTER
L2   A SIMPLE COUNTER
L3   A SIMPLE COUNTER
L4   A SIMPLE COUNTER
MAPST   THE SUBSCRIPT OF NBRAN SUCH THAT NBRAN IS THE BRANCH NUMBER IN ATTACK I4
MODE   AN OUTPUT CONTROL. IF 1 EFFECTIVENESS IS ARRANGED IN DESCENDING ORDER.
       IF 2 COST IS ARRANGED IN ASCENDING ORDER. IF 3 BOTH ARE DONE
MODE1      IF MODE EQUALS 1 OR 2 MODE1 EQUALS MODE. IF MODE EQUALS 3 MODE1 TAKES
           ON A VALUE OF FIRST 1 THEN 2 FOR USE IN EFFEC AND DELETE
MP   COUNTS THE NUMBER OF PERMUTATIONS WHICH ARE CURRENTLY BEING EXAMINED
MPER   IS A SENSE SWITCH. WHEN IT IS '0' THERE ARE NO PERMUTATIONS BEING EXAMINED
       IT IS '1' WHEN THERE ARE
MPER1   IS A SENSE SWITCH. '0' INDICATES THAT THE POSSIBILITY OF INVESTIGATING
        PERMUTATIONS DID NOT EXIST, AND '1' INDICATES THAT IT DID
```

MM COUNTS THE NUMBER OF PERMUTATIONS OF AN ATTACK ORDER FOR A SINGLE TARGET
M1 A SIMPLE COUNTER
M2 A SIMPLE COUNTER
M3 A SIMPLE COUNTER
M4 A SIMPLE COUNTER
M5 A SIMPLE COUNTER
M6 A SIMPLE COUNTER ASSUMING THE VALUES OF DIFFERENT ATTACK NUMBERS
M7 A SIMPLE COUNTER
M8 A SIMPLE COUNTER
M9 A SIMPLE COUNTER
NASGN THE LEAST NUMBER OF PLANES TO BE SENT
NASGN1 EQUALS NASGN MINUS 1
NM IF GREATER THAN 0, IT IS THE NEW VALUE OF SOME KAVL( )
NBRAN THE BRANCH NUMBER OF THE I3RD BRANCH READ INTO THE COMPUTER
NBRN TEMPORARY STORAGE OF BRANCH NUMBER
NBRNCH THE NUMBER OF BRANCHES TO BE EXAMINED
NPLANE THE NUMBER OF PLANES TO BE EXAMINED
NWEAP THE NUMBER OF WEAPON SYSTEMS TO BE EXAMINED
ORDER SUBROUTINE WHICH ARRANGES COST AND EFFECTIVENESS VALUES IN DESIRED ORDER
OUTPUT SUBROUTINE USED TO OUTPUT INFORMATION
PKL THE PROBABILITY OF KILL BY ALLIES ON FOE IN ATTACK NUMBER I4
PRCNT THE PROBABILITY OF KILL BY FOE ON ALLIES IN ATTACK NUMBER I4
PRCNT1 IS TEMPORARY STORAGE FOR PRCNT( )
PRCNT2 IS TEMPORARY STORAGE FOR PRCNT( )
READIN SUBROUTINE WHICH READS IN AND INITIALIZES
RATIO THE RATIO OF COST TO EFFECTIVENESS
STATE THE MAXIMUM AMOUNT OF SUPPLIES THAT MAY BE INTERDICTED IN THE I3RD BRANCH
BY ATTACKING DIFFERENT TARGETS. IF THE SECOND SUBSCRIPT IS 1, THE TARGET
IS THE STORAGE LOCATION. IF 2, THE VEHICLES. IF 3, THE ROUTE ITSELF
TOLC FOR DELETION PURPOSES, ATTACK COMBINATION COST VALUES DIFFERING BY LESS
THAN THIS VALUE ARE CONSIDERED TO BE THE SAME
TOLE FOR DELETION PURPOSES, ATTACK COMBINATION EFFECTIVENESS VALUES DIFFERING
BY LESS THAN THIS VALUE ARE CONSIDERED TO BE THE SAME
TRB THE TIME TO REBUILD A SPECIFIED BRANCH IF COMPLETELY DESTROYED
VCHEK TEMPORARY STORAGE OF VEHICLES EXPECTED ON A BRANCH
WGT( ) IS THE WEIGHTING FACTOR FOR THE I4TH ATTACK

```
K   THE VALUES OF L OR C IF MODE IS 1 OR 2 RESPECTIVELY
Y   THE VALUES OF C OR E IF MODE IS 1 OR 2 RESPECTIVELY
Y1   TEMPORARY STORAGE FOR Y
```

APPENDIX II

COMPUTER PROGRAM AND EXECUTION

```
      COMMON ISCL(100),KSOL(100),ITEST(10),IC(1000,10),KATK(500),
     *ISKIP(500),JSOL(100),CP(50),PKL(100),C(999),E(999),PRCNT(100),
     *KPTYPE(50),KOUT(300),KWSPL2(50),KASGN,KUT,KWSAVL(30),KPLAVL(30),JI
     *,NBRAN(50),STATE(50,3),MAPST(500),CM(50),HOLDST(50),HLDMAP(50),
     *NASGN,J1HOLD,J2HOLD,MODE,KSTOP,KSLHLD(10),NWEAP,I6END,TOLE,TOLC,
     *KWSPL(30),TRB(50),KWSASN(30),KPLASN(30),KWTYPE(50),J1OLD,CORD(100)
     *,IORDER(20),CPTEST(20),CPER(20) ,KAVL(100),IHDOR(20),IHDOR1(20),
     *EPER(20),WGT(100),IG(20),KBTR(100,20),KBT1(100),I4END,I4END1,IMDL,
     *NASGN1
      IMDL=0
      CALL READIN
    1 IF(KUT.EQ.1) CALL IMMOEL
      CALL COMBO
      IF(MODE.EQ.2) GOTO2
      MODE1=MODE
      IF(MODE.EQ.3) MODE1=1
      CALL ORDER(MODE1,J1HOLD,NASGN,E,C,IC)
      CALL OUTPUT
      IF(MODE.EQ.1) GOTO4
    2 MODE1=MODE
      IF(MODE.EQ.3) MODE1=2
      CALL ORDER (MODE1,J1HOLD,NASGN,C,E,IC)
    3 CALL OUTPUT
    4 NASGN=NASGN+1
      IF(NASGN.LE.KASGN) GOTO1
    5 STOP
      END
```

```
0001          SUBROUTINE READIN
0002          COMMON ISCL(100),KSOL(100),ITEST(10),IC(1000,10),KATK(500),
             *ISKIP(500),JSOL(100),CP(50),PKL(100),C(999),E(999),PRCNT(100),
             *KPTYPE(50),KOUT(300),KWSPL2(50),KASGN,KUT,KWSAVL(30),KPLAVL(30),J1
             *,NBRAN(50),STATE(50,3),MAPST(500),CM(50),HOLDST(50),HLDMAP(50),
             *NASGN,J1HOLD,J2HOLD,MODE,KSTOP,KSLHLD(10),NWEAP,I6END,TOLE,TOLC,
             *KWSPL(30),TRB(50),KWSASN(30),KPLASN(30),KWTYPE(50),J1OLD,CORD(100)
             *,IORDER(20),CPTEST(20),CPER(20) ,KAVL(100),IHDOR(20),IHDOR1(20),
             *EPER(20),WGT(100),IG(20),KBTR(100,20),KBT1(100),I4END,I4END1,IMDL,
             *NASGN1
0003          READ(5,101) NASGN,KASGN
0004          WRITE(6,102)NASGN,KASGN
0005          READ(5,117) MODE,KSTOP,TOLE,TOLC,KUT
0006          WRITE(6,118)MODE,KSTOP,TOLE,TOLC,KUT
0007          READ(5,103) NWEAP,NPLANE,NBRNCH
0008          WRITE(6,104)NWEAP,NPLANE,NBRNCH
0009          WRITE(6,105)
0010          I1=0
0011        1 I1=I1+1
0012          READ(5,106) KWTYP1,KWSAVL(I1),KWSPL(I1),CP(I1)
0013          WRITE(6,107)KWTYP1,KWSAVL(I1),KWSPL(I1),CP(I1)
0014          KWTYPE(I1)=KWTYP1
0015          IF(I1.LT.NWEAP) GOTO1
0016          WRITE(6,108)
0017          I20=0
0018      300 I20=I20+1
0019          KWSPL2(I20)=KWSPL(I20)
0020          IF(I20.LT.NWEAP) GOTO300
0021          I2=0
0022        2 I2=I2+1
```

```
      READ(5,109) KPTYP1,KPLAVL(I2)
      WRITE(6,110)KPTYP1,KPLAVL(I2)
      KPTYPE(I2)=KPTYP1
      I21=0
  301 I21=I21+1
      IF(KPTYP1.EQ.KWSPL2(I21)) KWSPL(I21)=I2
      IF(I21.LT.NWEAP) GOTO301
      IF(I2.LT.NPLANE) GOTO2
      WRITE(6,111)
      I3=0
    3 I3=I3+1
      READ(5,112) NBRN,VCHEX,AVCAP,AVFR,AVSPD,TRB(I3),STATE(I3,1)
      NBRAN(I3)=NBRN
      STATE(I3,2)=VCHEX*AVCAP
      STATE(I3,3)=TRB(I3)*AVFR*AVCAP
      WRITE(6,113)NBRN,VCHEX,AVCAP,AVFR,AVSPD,TRB(I3),STATE(I3,1),STATE(
     *I3,2),STATE(I3,3)
      IF(I3.LT.NBRNCH) GOTO3
      WRITE(6,114)
      I4=0
    4 I4=I4+1
    5 READ(5,115) ISOL1,JSOL(I4),KSOL(I4),PKL(I4),PRCNT(I4),CORD(I4),
     *WGT(I4)
      IF(ISOL1.LT.0) GOTO6
      I41=0
  501 I41=I41+1
      IF(JSOL(I4).EQ.NBRAN(I41)) GOTO502
      IF(I41.LT.NBRNCH) GOTO501
      WRITE(6,1151) JSOL(I4)
      STOP
  502 MAPST(I4)=I41
      I42=0
  503 I42=I42+1
      IF(ISOL1.EQ.KWTYPE(I42)) GOTO504
      IF(I42.LT.NWEAP) GOTO503
      WRITE(6,119) ISOL1
```

```
      STOP
  504 ISOL(I4)=I42
      KAVL(I4)=KWSAVL(I42)
      ISOL1=ISOL(I4)
      CM(I4)=CORD(I4)+CP(ISOL1)*PRCNT(I4)
      WRITE(6,116) KWTYPE(ISOL1),JSOL(I4),KSOL(I4),PKL(I4),PRCNT(I4),
     *CORD(I4),CP(ISOL1),CM(I4),WGT(I4)
      GOTO4
    6 I4END=I4-1
      I4END1=I4END-1
  101 FORMAT(2I5)
  102 FORMAT(1H1,' BELOW IS THE DATA INPUT'//' MINIMUM NUMBER TO BE SENT
     * IS',I5/' MAXIMUM NUMBER TO BE SENT IS', I5/)
  103 FORMAT(3I5)
  104 FORMAT(' NUMBER OF DIFFERENT TYPES OF WEAPON SYSTEMS IS',I5/
     *' NUMBER OF DIFFERENT TYPES OF PLANES IS',I5/' NUMBER OF BRANCHES
     * IS',I5////)
  105 FORMAT(' W/S TYPE    AVAILABILITY    PLANE TYPE USED    PLANE COST'//
     *)
  106 FORMAT(3I5, F10.0)
  107 FORMAT(I7,I17,I16,F19.0)
  108 FORMAT(///' PLANE TYPE    AVAILABILITY'//)
  109 FORMAT(2I5)
  110 FORMAT(I11,I8)
  111 FORMAT(////' STATE 1 REFERS TO POTENTIAL INTERDICTION AT TARGET 1
     *(SUPPLIES)'/' STATE 2 REFERS TO POTENTIAL INTERDICTION AT TARGET 2
     * (VEHICLES)'/' STATE 3 REFERS TO POTENTIAL INTERDICTION AT TARGET
     *3 (ROUTE)'//' BRANCH  VEH EXPT  AVG LBS  VEH/HR  AVG MPH  HRS TO R
     *BLD  STATE1  STATE 2  STATE 3'//)
  112 FORMAT(I10,6F10.0)
  113 FORMAT(I7,F10.2,F9.2,F8.2,F9.2,F13.2,3F9.2)
  114 FORMAT(////'     W/S  BRANCH  TARGET  PK ON FOE  PK ON ALLY  VAR CO
     *ST  W/S COST  MISSION COST  WEIGHT'//)
  115 FORMAT(3I5,4F10.0)
 1151 FORMAT(///' PROGRAM TERMINATED. INPUT DATA INDICATES AN ATTACK POS
     *SIBILITY EXISTS FOR BRANCH',I5,',','/' BUT NO DATA HAS BEEN PROVIDED
```

```
      * FOR THIS BRANCH')
  116 FORMAT(I7,2I8,F11.2,F12.2,2F10.2,F14.2,F8.2)
  117 FORMAT(2I5,2F10.0,I5)
  118 FORMAT(//' OUTPUT MODE IS',I5/'      IF MODE IS 1, ARRANGE EFFECTIV
     *ENESS VALUES IN DESCENDING ORDER'/ '      IF MODE IS 2 ARRANGE COST
     * VALUES IN ASCENDING ORDER'/'      IF MODE IS 3 PROCEDE AS IF MODE
     *WERE 1, THEN 2'/' OUTPUT CONTROL FOR NUMBER OF ATTACK COMBINATIONS
     * OUTPUT IS',I5/'      A VALUE OF 0 INDICATES ALL COMBINATIONS ARE T
     *O BE OUTPUT'/'      ANY OTHER VALUE IS THE NUMBER OF ATTACK COMBINA
     *TIONS TO BE OUTPUT'/' ANY EFFECTIVENESS VALUES DIFFERING BY LESS T
     *HAN',F10.2,' ARE CONSIDERED TO BE THE SAME'/' ANY COST VALUES DIFF
     *ERING BY LESS THAN',F10.2,' ARE CONSIDERED TO BE THE SAME'/' THE D
     *ELETION CONTROL IS',I2,'. IF 1, ATTACK COMBINATIONS WITH A LOWER V
     *ALUE OF EFFECTIVENESS.'/' BUT A HIGHER COST THEN SOME OTHER ATTACK
     * COMBINATION, ARE DELETED.'/' IF 0, THEY ARE NOT.'//)
  119 FORMAT(' ERROR. NO INFORMATICN ABOUT W/S ',I5,' HAS BEEN READ IN,
     *BUT AN ATTACK USES THIS WEAPON SYSTEM'/' PROGRAM STOPPED')
      RETURN
      END
```

```
      SUBROUTINE IMMDEL
      COMMON ISOL(100),KSOL(100),ITEST(10),IC(1000,10),KATK(500),
     *ISKIP(500),JSOL(100),CP(50),PKL(100),C(999),E(999),PRCNT(100),
     *KPTYPE(50),KOUT(300),KWSPL2(50),KASGN,KUT,KWSAVL(30),KPLAVL(30),J1
     *,NBRAN(50),STATE(50,3),MAPST(5CO),CM(50),HOLDST(50),HLDMAP(50),
     *NASGN,J1HOLD,J2HOLD,MODE,KSTOP,KSLHLD(10),NWEAP,I6END,TOLF,TOLC,
     *KWSPL(30),TRB(50),KWSASN(30),KPLASN(30),KWTYPE(50),J1OLD,CORD(100)
     *,IORDER(20),CPTEST(20),CPER(20) ,KAVL(100),IHDOR(20),IHDOR1(20),
     *EPER(20),WGT(100),IG(20),KBTR(100,20),KBT1(100),I4END,I4END1,IMDL,
     *NASGN1
      IF(IMDL.EQ.0) GOTO30
      I451=0
  300 I451=I451+1
      ISOL1=ISOL(I451)
      KAVL(I451)=KWSAVL(ISOL1)
      IF(I451.LT.I4END) GOTO3CO
      GOTO181
   30 I45=0
   31 I45=I45+1
      KBT1(I45)=0
      IF(I45.LT.I4END) GOTO31
      I5=0
    1 I5=I5+1
      I6=I5
    2 I6=I6+1
      IF(JSOL(I5).NE.JSOL(I6)) GOTO18
      IF(KSOL(I5).NE.KSOL(I6)) GOTO18
      IF(PKL(I5)-PKL(I6)) 14,4,10
    4 IF(CM(I5)-CM(I6)) 8,5,9
    5 IF(CORD(I5)-CORD(I6)) 6,18,7
    6 KBT1(I6)=KBT1(I6)+1
      KBT=KBT1(I6)
      KBTR(I6,KBT)=I5
      GOTO18
    7 KBT1(I5)=KBT1(I5)+1
      KBT=KBT1(I5)
```

```
      KBTR(I5,KBT)=I6
      GOTO18
    8 IF(CORD(I5).GT.CORD(I6)) GOTO18
      KBT1(I6)=KBT1(I6)+1
      KBT=KBT1(I6)
      KBTR(I6,KBT)=I5
      GOTO18
    9 IF(CORD(I6).GT.CORD(I5)) GOTO18
      KBT1(I5)=KBT1(I5)+1
      KBT=KBT1(I5)
      KBTR(I5,KBT)=I6
      GOTO18
   10 IF(CM(I5).GT.CM(I6)) GOTO18
      IF(CORD(I5).GT.CORD(I6)) GOTO18
      KBT1(I6)=KBT1(I6)+1
      KBT=KBT1(I6)
      KBTR(I6,KBT)=I5
      GOTO18
   14 IF(CM(I6).GT.CM(I5)) GOTO18
      IF(CORD(I6).GT.CORD(I5)) GOTO18
      KBT1(I5)=KBT1(I5)+1
      KBT=KBT1(I5)
      KBTR(I5,KBT)=I6
   18 IF(I6.LT.I4END) GOTO2
      IF(I5.LT.I4END1) GOTO1
      IMDL=1
  181 I55=0
   19 I55=I55+1
      I56=0
      KB=0
   20 I56=I56+1
      IF(I56.GT.KBT1(I55)) GOTO21
      KB1=KBTR(I55,I56)
      KB=KB+KAVL(KB1)
      GOTO20
   21 NB=NASGN-KB
```

```
      IF(KAVL(I55).GT.NB) KAVL(I55)=NB
      IF(KAVL(I55).LT.0) KAVL(I55)=0
      IF(I55.LT.I4END) GOTO19
      I51=0
      I61=0
   22 I51=I51+1
      IF(KAVL(I51).EQ.0) GOTO24
      I71=0
      ISKIP1=I61+1+KAVL(I51)
   23 I71=I71+1
      I61=I61+1
      KATK(I61)=I51
      ISKIP(I61)=ISKIP1
      IF(I71.LT.KAVL(I51)) GOTO23
   24 IF(I51.LT.I4END) GOTO22
      I6END=I61
      I6END1=I6END-1
      RETURN
      END
```

```
      SUBROUTINE COMBO
      COMMON ISOL(100),KSOL(100),ITEST(10),IC(1000,10),KATK(500),
     *ISKIP(500),JSOL(100),CP(50),PKL(100),C(999),E(999),PRCNT(100),
     *KPTYPE(50),KOUT(300),KWSPL2(50),KASGN,KUT,KWSAVL(30),KPLAVL(30),J1
     *,NBRAN(50),STATE(50,3),MAPST(500),CM(50),HOLDST(50),HLDMAP(50),
     *NASGN,J1HOLD,J2HOLD,MODE,KSTOP,KSLHLD(10),NWEAP,I6END,TOLE,TOLC,
     *KWSPL(30),TRB(50),KWSASN(30),KPLASN(30),KWTYPE(50),J1OLD,CORD(100)
     *,IORDER(20),CPTEST(20),CPER(20) ,KAVL(100),IHDOR(20),IHDOR1(20),
     *EPER(20),WGT(100),IG(20),KBTR(100,20),KBT1(100),I4END,I4END1,IMDL,
     *NASGN1
      NASGN1=NASGN-1
      I8=0
    9 I8=I8+1
      KWSASN(I8)=0
      KPLASN(I8)=0
      IF(I8.LT.NWEAP)GOTO9
      I9=0
   10 I9=I9+1
      ITEST(I9)=I9
      IF(I9.LT.NASGN)GOTO10
      J1=0
      J2=0
   11 J2=J2+1
   12 ITEST1=ITEST(J2)
      KATK1=KATK(ITEST1)
      ISOL1=ISOL(KATK1)
      KWSPL1=KWSPL(ISOL1)
      KWSASN(ISOL1)=KWSASN(ISOL1)+1
      KPLASN(KWSPL1)=KPLASN(KWSPL1)+1
      IF(KPLASN(KWSPL1).GT.KPLAVL(KWSPL1)) GOTO13
      IF(KWSASN(ISOL1).LE.KWSAVL(ISOL1)) GOTO17
   13 J2HOLD=J2
      KWSASN(ISOL1)=KWSASN(ISOL1)-1
      KPLASN(KWSPL1)=KPLASN(KWSPL1)-1
      KHECK=I6END-(NASGN-J2)
      IF(ISKIP(ITEST1).LE.KHECK) GOTO14
```

```
      J2=J2-1
      IF(J2.EQ.0) GOTO19
      ITEST1=ITEST(J2)
      KATK1=KATK(ITEST1)
      ISOL1=ISOL(KATK1)
      KWSPL1=KWSPL(ISOL1)
      GOTO13
   14 ITEST(J2)=ISKIP(ITEST1)
   15 J2=J2+1
      ITEST(J2)=ITEST(J2-1)+1
      IF(J2.LT.NASGN) GOTO15
   16 J2=J2HOLD
      GOTO12
   17 IF(J2.LT.NASGN) GOTO11
      J1=J1+1
      J1HOLD=J1
      J1OLD=J1
      J3=0
   18 J3=J3+1
      IC(J1,J3)=ITEST(J3)
      IF(J3.LT.NASGN) GOTO18
      CALL EFFEC
      IF(KUT.EQ.1) CALL DELETE
      GOTO13
   19 RETURN
      END
```

```
0001          SUBROUTINE EFFEC
0002          COMMON ISOL(100),KSOL(100),ITEST(10),IC(1000,10),KATK(500),
             *ISKIP(500),JSOL(100),CP(50),PKL(100),C(999),E(999),PRCNT(100),
             *KPTYPE(50),KOUT(300),KWSPL2(50),KASGN,KUT,KWSAVL(30),KPLAVL(30),J1
             *,NBRAN(50),STATE(50,3),MAPST(500),CM(50),HOLDST(50),HLDMAP(50),
             *NASGN,J1HOLD,J2HOLD,MODE,KSTOP,KSLHLD(10),NWEAP,I6END,TOLE,TOLC,
             *KWSPL(30),TRB(50),KWSASN(30),KPLASN(30),KWTYPE(50),J1OLD,CORD(100)
             *,IORDER(20),CPTEST(20),CPFR(20) ,KAVL(100),IHDOR(20),IHDOR1(20),
             *EPER(20),WGT(100),IG(20),KBTR(100,20),KBT1(100),I4END,I4END1,IMDL,
             *NASGN1
0003          MR=0
0004          MP=0
0005      100 K40=0
0006      101 K40=K40+1
0007          IC1=IC(J1HOLD,K40)
0008          IORDER(K40)=IC1
0009          IF(K40.LT.NASGN) GOTO101
0010          K43=0
0011      102 K41=K43
0012      103 MPER=0
0013          MP=0
0014          MPER1=0
0015      104 K41=K41+1
0016          IF(K41.GT.NASGN) GOTO361
0017          IF(K41.GT.NASGN1) GOTO142
0018          K43=K41
0019          IC1=IORDER(K41)
0020          KATK1=KATK(IC1)
0021          K42=K41
0022      105 K42=K42+1
0023          IC2=IORDER(K42)
0024          KATK2=KATK(IC2)
0025          IF(JSOL(KATK1).NE.JSOL(KATK2)) GOTO106
0026          IF(KSOL(KATK1).NE.KSOL(KATK2)) GOTO106
0027          MPER=1
0028          MPER1=1
```

```
      K43=K43+1
      IODER1=IORDER(K43)
      IORDER(K43)=IORDER(K42)
      IORDER(K42)=IODER1
  106 IF(K42.LT.NASGN) GOTO105
      IF(MPER.EQ.0) GOTO142
  107 K4=K41-1
   11 K4=K4+1
      IODER1=IORDER(K4)
      KATK1=KATK(IODER1)
      ISOL1=ISOL(KATK1)
      CPTEST(K4)=CP(ISOL1)
      IF(K4.LT.K43) GOTO11
      K5=K41-1
   12 K5=K5+1
      K6=K5
   13 K6=K6+1
      IF(K6.GT.K43) GOTO14
      IF(CPTEST(K5).LE.CPTEST(K6)) GOTO14
      CPTST1=CPTEST(K5)
      CPTEST(K5)=CPTEST(K6)
      CPTEST(K6)=CPTST1
      IODER1=IORDER(K5)
      IORDER(K5)=IORDER(K6)
      IORDER(K6)=IODER1
      GOTO13
   14 CPX=1.
      IODER2=IORDER(K41)
      PRCNT2=PRCNT(KATK2)
      KATK2=KATK(IODER2)
      K61=K41-1
  141 K61=K61+1
      IODER1=IODER2
      PRCNT1=PRCNT2
      IODER2=IORDER(K61)
      KATK2=KATK(IODER2)
```

```
      CPX=CPX*(1.-PKL(KATK1))
      PRCNT2=PRCNT(KATK2)*CPX
      IF(PRCNT2.GT.PRCNT1) GOTO142
      IF(K61.LT.K43) GOTO141
      MPER=0
  142 CPER1=0.
      EPER1=0.
      K1=K41-1
      K11=0
    1 K1=K1+1
      K11=K11+1
      IODER1=IORDER(K1)
      KATK1=KATK(IODER1)
      KSOL1=KSOL(KATK1)
      MAPST1=MAPST(KATK1)
      HOLDST(K11)=STATE(MAPST1,KSOL1)
      IF(K1.LT.K43) GOTO1
      K2=K41-1
      K20=0
    2 K2=K2+1
      K20=K20+1
      IODER1=IORDER(K2)
      KATK1=KATK(IODER1)
      KSOL1=KSOL(KATK1)
      MAPST1=MAPST(KATK1)
      ISOL1=ISOL(KATK1)
      HIT=STATE(MAPST1,KSOL1)*PKL(KATK1)
      EPER1=EPER1+HIT*WGT(KATK1)
      CPER1=CPER1+CORD(ISOL1)+CP(ISOL1)*PRCNT(KATK1)*(STATE(MAPST1,KSOL1
     *)/HOLDST(K20))
      STATE(MAPST1,KSOL1)=STATE(MAPST1,KSOL1)-HIT
      IF(K2.LT.K43) GOTO2
      K3=K41-1
      K30=0
    3 K3=K3+1
      K30=K30+1
```

```
      IODER1=IORDER(K3)
      KATK1=KATK(IODER1)
      MAPST1=MAPST(KATK1)
      KSOL1=KSOL(KATK1)
      STATE(MAPST1,KSOL1)=HOLDST(K30)
      IF(K3.LT.K43) GOTO3
      IF(MPER.EQ.1) GOTO20
   36 MR=MR+1
      EPER(MR)=EPER1
      CPER(MR)=CPER1
      IF(K43.GT.NASGN1) GOTO361
      IF(MPER1.EQ.1) GOTO0102
      IF(NASGN.EQ.1) GOTO0361
      GOTO0104
   20 MP=MP+1
      IF(MP.GT.1) GOTO3600
      K31=K41-1
  360 K31=K31+1
      IHDOR(K31)=IORDER(K31)
      IF(K31.LT.K43) GOTO0360
      MR=MR+1
      CPER(MR)=CPER1
      EPER(MR)=EPER1
      COLD=CPER1
      GOTO21
 3600 IF(COLD.LT.CPER1) GOTO0301
      COLD=CPER1
      CPER(MR)=COLD
      K32=K41-1
 3601 K32=K32+1
      IHDOR(K32)=IORDER(K32)
      IF(K32.LT.K43) GOTO03601
      GOTO0301
   21 K44END=K43-K41+1
      K44=0
   30 K44=K44+1
```

```
      IG(K44)=K44
      IF(K44.LT.K44END) GOTO30
      K45=K44END
301   K45=K45-1
      IF(K45.EQ.0) GOTO310
      K46=IG(K45)
      IF(K46.EQ.K44END) GOTO301
302   K46=K46+1
      IF(K46.GT.K44END) GOTO301
      K47=0
303   K47=K47+1
      IF(IG(K47).EQ.K46) GOTO302
      IF(K47.LT.K45) GOTO303
      IG(K45)=K46
      K48=0
304   K48=K48+1
      K481=0
305   K481=K481+1
      IF(K481.GT.K45) GOTO306
      IF( IG(K481).EQ.K48) GOTO304
      GOTO305
306   K45=K45+1
      IG(K45)=K48
      IF(K45.LT.K44END) GOTO304
307   K49=K41-1
308   K49=K49+1
      IHDOR1(K49)=IORDER(K49)
      IF(K49.LT.K43) GOTO308
      K491=0
      K492=K41-1
309   K491=K491+1
      K492=K492+1
      K493=K41+IG(K491)-1
      IORDER(K492)=IHDOR1(K493)
      IF(K492.LT.K43) GOTO309
      GOTO142
```

```
  310 K494=K41-1
  311 K494=K494+1
      IORDER(K494)=IHDOR(K494)
      IF(K494.LT.K43) GOTO311
      IF(K43.LT.NASGN) GOTO102
  361 C(J1HOLD)=0.
      E(J1HOLD)=0.
 3610 K50=0
  362 K50=K50+1
      C(J1HOLD)=C(J1HOLD)+CPER(K50)
      E(J1HOLD)=E(J1HOLD)+EPER(K50)
      IF(K50.LT.MR) GOTO362
      K51=0
  363 K51=K51+1
      IC(J1HOLD,K51)=IORDER(K51)
      IF(K51.LT.NASGN) GOTO363
      RETURN
      END
```

```
      SUBROUTINE DELETE
      COMMON ISCL(100),KSOL(100),ITEST(10),IC(1000,10),KATK(500),
     *ISKIP(500),JSOL(100),CP(5C),PKL(100),C(999),E(999),PRCNT(100),
     *KPTYPE(50),KOUT(300),KWSPL2(50),KASGN,KUT,KWSAVL(30),KPLAVL(30),J1
     *,NBRAN(50),STATE(50,3),MAPST(500),CM(50),HOLDST(50),HLDMAP(50),
     *NASGN,J1HOLD,J2HOLD,MODE,KSTOP,KSLHLD(10),NWEAP,I6END,TOLE,TOLC,
     *KWSPL(30),TRB(50),KWSASN(30),KPLASN(30),KWTYPE(50),J1OLD,CORD(100)
     *,IORDER(20),CPTEST(20),CPER(20) ,KAVL(100),IHDOR(20),IHDOR1(20),
     *EPER(20),WGT(10C),IG(20),KBTR(100,20),KBT1(100),I4END,I4END1,IMDL,
     *NASGN1
      K4=0
      K5=0
      ETEST=E(J1HOLD)
      CTEST=C(J1HOLD)
      J1HLD1=J1HOLD-1
    1 K4=K4+1
      IF(K4.GT.J1HLD1) GOTO4
      IF(ETEST.LT.E(K4)) GOTO2
      GOTO3
    2 IF(CTEST.LE.C(K4)) GOTO1
      K5=K5+1
      KOUT(K5)=J1HOLD
      GOTO4
    3 IF(CTEST.GE.C(K4)) GOTO1
      K5=K5+1
      KOUT(K5)=K4
      GOTO1
    4 IF(K5.EQ.C) GOTO9
      K6=0
      K6STOP=K5
    5 K6=K6+1
      K7=KOUT(K6)-K6
      K7STOP=J1HOLD-K6
      IF(KOUT(K6).EQ.J1HOLD) GOTO8
    6 K7=K7+1
      K8=K7+1
```

```
      E(K7)=F(K8)
      C(K7)=C(K8)
      K9=0
    7 K9=K9+1
      IC(K7,K9)=IC(K8,K9)
      IF(K9.LT.NASGN) GOTO7
      IF(K7.LT.K7STOP) GOTO6
      IF(K6.LT.K6STOP) GOTO5
    8 J1=K7STOP
      J1HOLD=K7STOP
    9 IF(NASGN.NE.6) RETURN
      RETURN
      END
```

```
      SUBROUTINE ORDER(MODE,J1HOLD,NASGN,X,Y,IC)
      DIMENSION X(500),Y(500),IC(1000,10)
      J1HLD1=J1HOLD-1
      K4=0
    1 K4=K4+1
      K5=K4
    2 K5=K5+1
      IF(MODE.EQ.1.AND.X(K4).GT.X(K5)) GOTO4
      IF(MODE.EQ.2.AND.X(K4).LT.X(K5)) GOTO4
      X1=X(K4)
      X(K4)=X(K5)
      X(K5)=X1
      Y1=Y(K4)
      Y(K4)=Y(K5)
      Y(K5)=Y1
      K6=0
    3 K6=K6+1
      IC1=IC(K4,K6)
      IC(K4,K6)=IC(K5,K6)
      IC(K5,K6)=IC1
      IF(K6.LT.NASGN) GOTO3
    4 IF(K5.LT.J1HOLD) GOTO2
      IF(K4.LT.J1HLD1) GOTO1
      RETURN
      END
```

```
      SUBROUTINE OUTPUT
      COMMON ISOL(100),KSOL(100),ITEST(10),IC(1000,10),KATK(500),
     *ISKIP(500),JSOL(100),CP(50),PKL(100),C(999),E(999),PRCNT(100),
     *KPTYPE(50),KOUT(300),KWSPL2(50),KASGN,KUT,KWSAVL(30),KPLAVL(30),J1
     *,NBRAN(50),STATE(50,3),MAPST(500),CM(50),HOLDST(50),HLDMAP(50),
     *NASGN,J1HOLD,J2HOLD,MODE,KSTOP,KSLHLD(10),NWEAP,I6END,TOLE,TOLC,
     *KWSPL(30),TRB(50),KWSASN(30),KPLASN(30),KWTYPE(50),J1OLD,CORD(100)
     *,IORDER(20),CPTEST(20),CPER(20) ,KAVL(100),IHDOR(20),IHDOR1(20),
     *EPER(20),WGT(100),IG(20),KBTR(100,20),KBT1(100),I4END,I4END1,IMDL,
     *NASGN1
      IF(KSTOP.GT.0.AND.J1HOLD.GT.KSTOP) J1HOLD=KSTOP
      WRITE(6,101) NASGN
      L2=0
    1 L2=L2+1
      L3=0
    2 L3=L3+1
      IC1=IC(L2,L3)
      KATK1=KATK(IC1)
      ISOL1=ISOL(KATK1)
      JSOL1=JSOL(KATK1)
      MAPST1=MAPST(KATK1)
      KSOL1=KSOL(KATK1)
      IF(KSOL1.EQ.3) GOTO20
      WRITE(6,102) L2,KWTYPE(ISOL1),JSOL1,KSOL1,STATE(MAPST1,KSOL1)
      GOTO21
   20 WRITE(6,105) L2,KWTYPE(ISOL1),JSOL1,KSOL1,STATE(MAPST1,KSOL1),
     *TRB(JSOL1)
   21 IF(L3.LT.NASGN) GOTO2
      IF(L2.LT.J1HOLD) GOTO1
      WRITE(6,103) NASGN
      L4=0
    3 L4=L4+1
      RATIO=C(L4)/E(L4)
      WRITE(6,104) L4,C(L4),E(L4),RATIO
      IF(L4.LT.J1HOLD) GOTO3
  101 FORMAT(1H1///////////////////////////////' OUTPUT OF INFORMATION FOR
```

```
     * ATTACK COMBINATIONS OF' I5,' WEAPON SYSTEMS'/1H1//' ATTACK NUMBER
     *    W/S    BRANCH    TARGET    INITIAL STATE    HOURS TO REBUILD'////)
 102 FORMAT(I14,I6,I9,I9,F15.2,'                    ---'/)
 103 FORMAT(1H1////////////////////////////////////' SUMMARY INFORMATION FOR A
     *TTACK COMBINATIONS OF' I5,' WEAPON SYSTEMS'/1H1,' ATTACK COMBINATI
     *ON   DOLLARS COST   EFFECTIVENESS IN LBS.   COST/EFFECTIVENESS'//)
 104 FORMAT(I19,F15.2,F23.2,F21.2)
 105 FORMAT(I14,I6,I9,I9,F15.2,F16.1)
     RETURN
     END
```

```
BELOW IS THE DATA INPUT

MINIMUM NUMBER TO BE SENT IS     1
MAXIMUM NUMBER TO BE SENT IS     4


OUTPUT MODE IS     1
    IF MODE IS 1, ARRANGE EFFECTIVENESS VALUES IN DESCENDING ORDER
    IF MODE IS 2 ARRANGE COST VALUES IN ASCENDING ORDER
    IF MODE IS 3 PROCEDE AS IF MODE WERE 1, THEN 2
OUTPUT CONTROL FOR NUMBER OF ATTACK COMBINATIONS OUTPUT IS     0
    A VALUE OF 0 INDICATES ALL COMBINATIONS ARE TO BE OUTPUT
    ANY OTHER VALUE IS THE NUMBER OF ATTACK COMBINATIONS TO BE OUTPUT
ANY EFFECTIVENESS VALUES DIFFERING BY LESS THAN       0.0  ARE CONSIDERED TO BE THE SAME
ANY COST VALUES DIFFERING BY LESS THAN       0.0  ARE CONSIDERED TO BE THE SAME
THE DELETION CONTROL IS 1. IF 1, ATTACK COMBINATIONS WITH A LOWER VALUE OF EFFECTIVENESS,
BUT A HIGHER COST THEN SOME OTHER ATTACK COMBINATION, ARE DELETED.
IF 0, THEY ARE NOT.


NUMBER OF DIFFERENT TYPES OF WEAPON SYSTEMS IS     3
NUMBER OF DIFFERENT TYPES OF PLANES IS     2
NUMBER OF BRANCHES  IS     2



W/S TYPE   AVAILABILITY   PLANE TYPE USED   PLANE COST


    11               2                1             800000.
    12               2                1             800000.
    18               2                2             900000.
```

PLANE TYPE	AVAILABILITY
1	2
2	2

STATE 1 REFERS TO POTENTIAL INTERDICTION AT TARGET 1 (SUPPLIES)
STATE 2 REFERS TO POTENTIAL INTERDICTION AT TARGET 2 (VEHICLES)
STATE 3 REFERS TO POTENTIAL INTERDICTION AT TARGET 3 (ROUTE)

BRANCH	VEH EXPT	AVG LBS	VEH/HR	AVG MPH	HRS TO RBLD	STATE1	STATE 2	STATE 3
1	2.00	100.00	10.00	20.00	30.00	50.00	200.00	30000.00
2	3.00	10.00	5.00	15.00	20.00	70.00	210.00	11200.00

W/S	BRANCH	TARGET	PK ON FOF	PK ON ALLY	VAR COST	W/S COST	MISSION COST	WEIGHT
11	1	1	0.30	0.40	90000.00	800000.00	409999.94	1.00
12	1	1	0.15	0.35	100000.00	800000.00	379999.94	1.00
18	1	1	0.18	0.30	120000.00	900000.00	389999.94	1.00
11	2	3	0.05	0.20	90000.00	800000.00	249999.94	1.00
12	2	3	0.10	0.15	100000.00	800000.00	219999.94	1.00
18	2	3	0.15	0.12	120000.00	900000.00	227999.94	1.00
11	2	2	0.20	0.15	90000.00	800000.00	209999.94	1.00
12	2	2	0.15	0.20	100000.00	800000.00	259999.94	1.00
18	2	2	0.10	0.18	120000.00	900000.00	281999.94	1.00

OUTPUT OF INFORMATION FOR ATTACK COMBINATIONS OF 1 WEAPON SYSTEMS

ATTACK NUMBER	W/S	BRANCH	TARGET	INITIAL STATE	HOURS TC REBUILD
1	18	2	3	11200.00	20.0
2	12	2	3	11200.00	20.0
3	11	2	2	210.00	---

SUMMARY INFORMATION FOR ATTACK COMBINATIONS OF 1 WEAPON SYSTEMS

ATTACK COMBINATION	COLLARS COST	EFFECTIVENESS IN LBS.	COST/EFFECTIVENESS
1	227999.94	1680.00	135.71
2	219999.94	1120.00	196.43
3	209999.94	42.00	5000.00

OUTPUT OF INFORMATION FOR ATTACK COMBINATIONS OF 2 WEAPON SYSTEMS

ATTACK NUMBER	W/S	BRANCH	TARGET	INITIAL STATE	HOURS TO REBUILD
1	18	2	3	11200.00	20.0
1	18	2	3	11200.00	20.0
2	18	2	3	11200.00	20.0
2	12	2	3	11200.00	20.0
3	12	2	3	11200.00	20.0
3	12	2	3	11200.00	20.0
4	11	2	2	210.00	---
4	11	2	2	210.00	---

SUMMARY INFORMATION FOR ATTACK COMBINATIONS OF 2 WEAPON SYSTEMS

ATTACK COMBINATION	DOLLARS COST	EFFECTIVENESS IN LBS.	COST/EFFECTIVENESS
1	439799.88	3108.00	141.51
2	429999.88	2632.00	163.37
3	427999.68	2128.00	201.13
4	395999.88	75.60	5238.09

OUTPUT OF INFORMATION FOR ATTACK COMBINATIONS OF 3 WEAPON SYSTEMS

ATTACK NUMBER	W/S	BRANCH	TARGET	INITIAL STATE	HOURS TO REBUILD
1	18	2	3	11200.00	20.0
1	12	2	3	11200.00	20.0
1	18	2	3	11200.00	20.0
2	18	2	3	11200.00	20.0
2	12	2	3	11200.00	20.0
2	12	2	3	11200.00	20.0

SUMMARY INFORMATION FOR ATTACK COMBINATIONS OF 3 WEAPON SYSTEMS

ATTACK COMBINATION	DOLLARS COST	EFFECTIVENESS IN LBS.	COST/EFFECTIVENESS
1	632619.81	3917.20	161.50
2	621799.81	3488.80	178.23

OUTPUT OF INFORMATION FOR ATTACK COMBINATIONS OF 4 WEAPON SYSTEMS

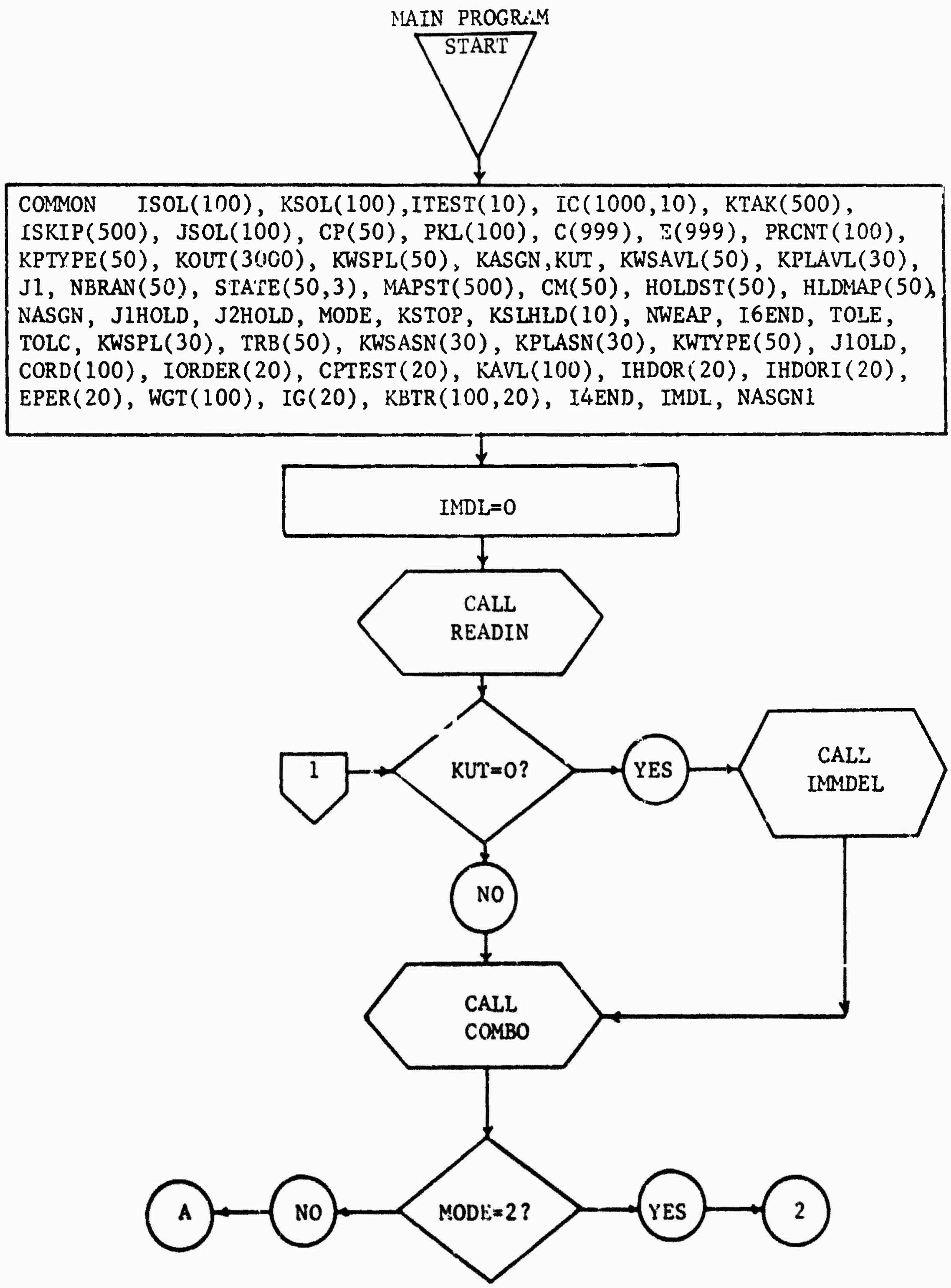
ATTACK NUMBER	W/S	BRANCH	TARGET	INITIAL STATE	HOURS TO REBUILD
1	18	2	3	11200.00	20.0
1	18	2	3	11200.00	20.0
1	12	2	3	11200.00	20.0
1	12	2	3	11200.00	20.0

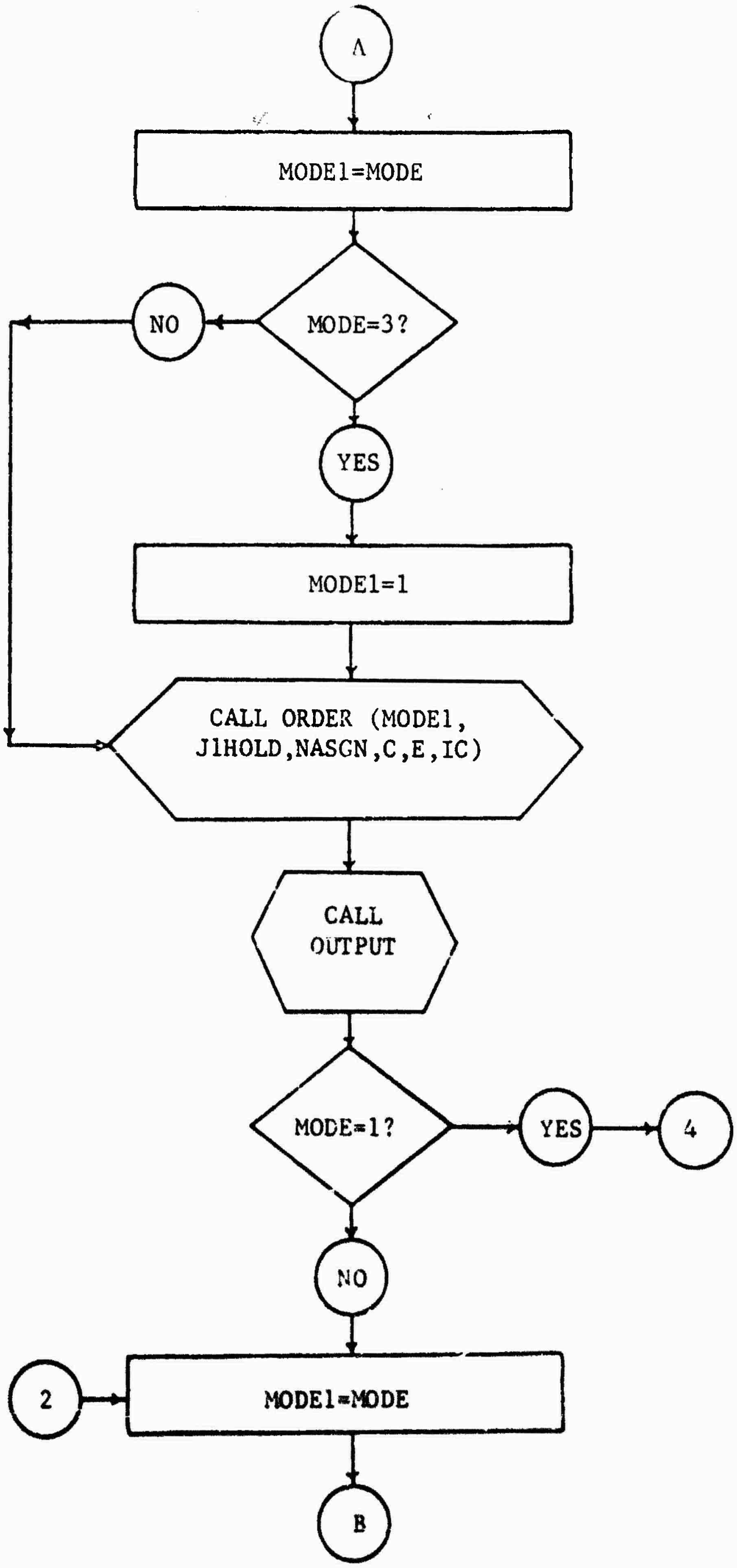
SUMMARY INFORMATION FOR ATTACK COMBINATIONS OF 4 WEAPON SYSTEMS

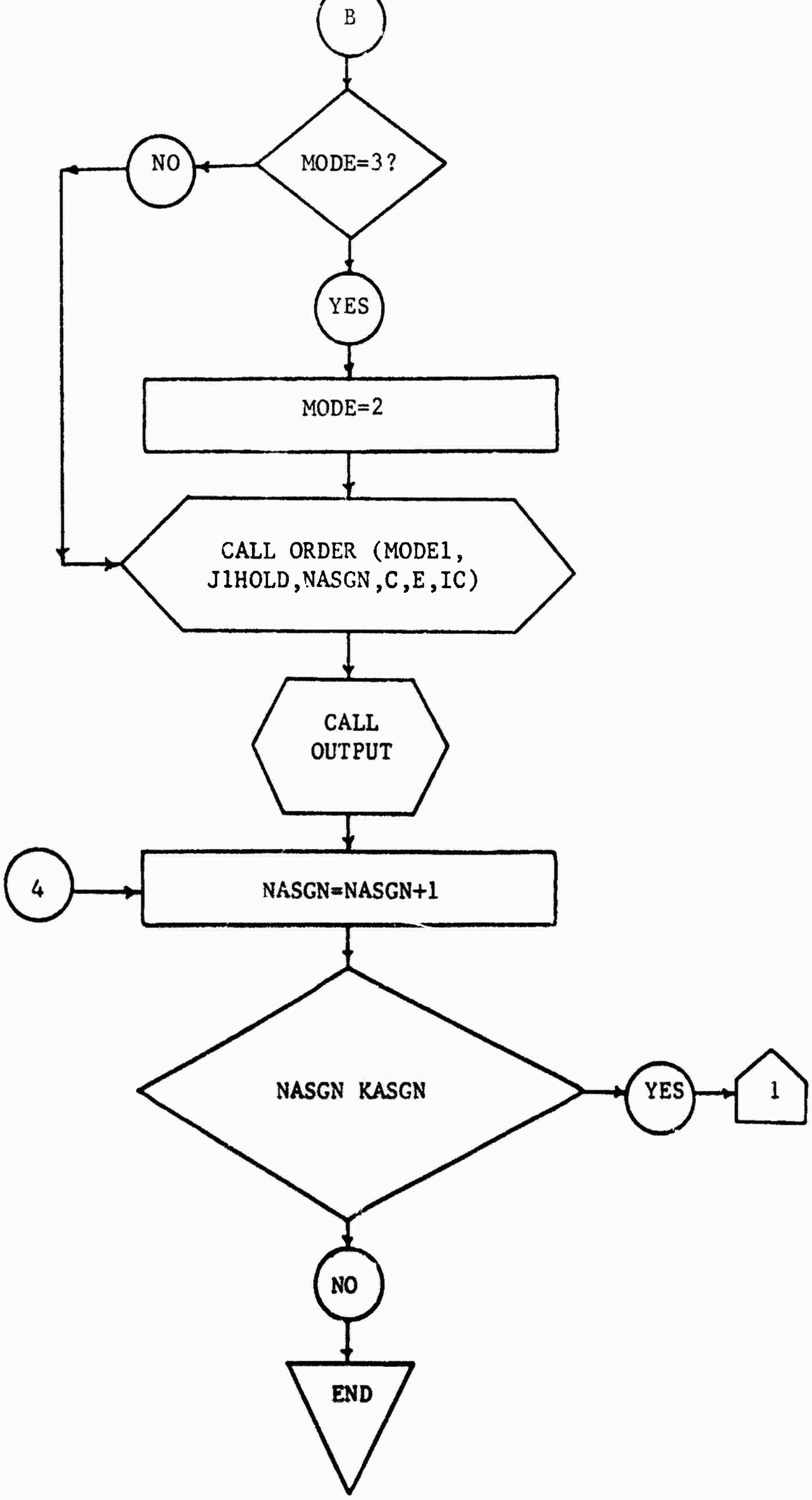
ATTACK COMBINATION	DOLLARS COST	EFFECTIVENESS IN LBS.	COST/EFFECTIVENESS
1	804529.69	4645.48	173.19

APPENDIX III

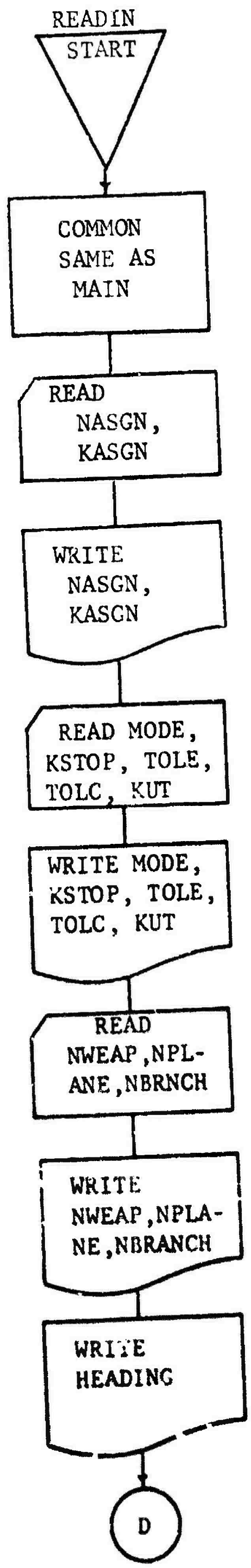
MICRO-LOGIC FLOW CHART OF COMPUTER MODEL

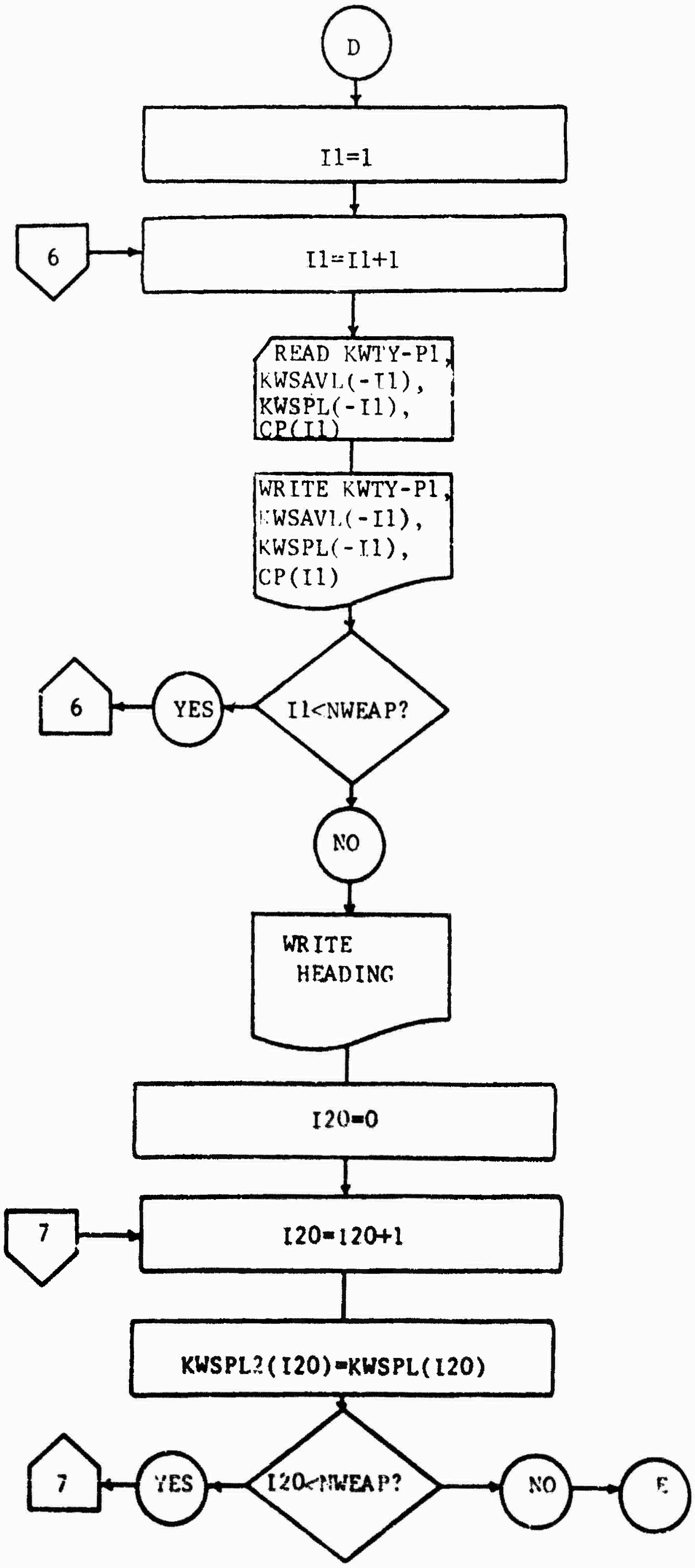


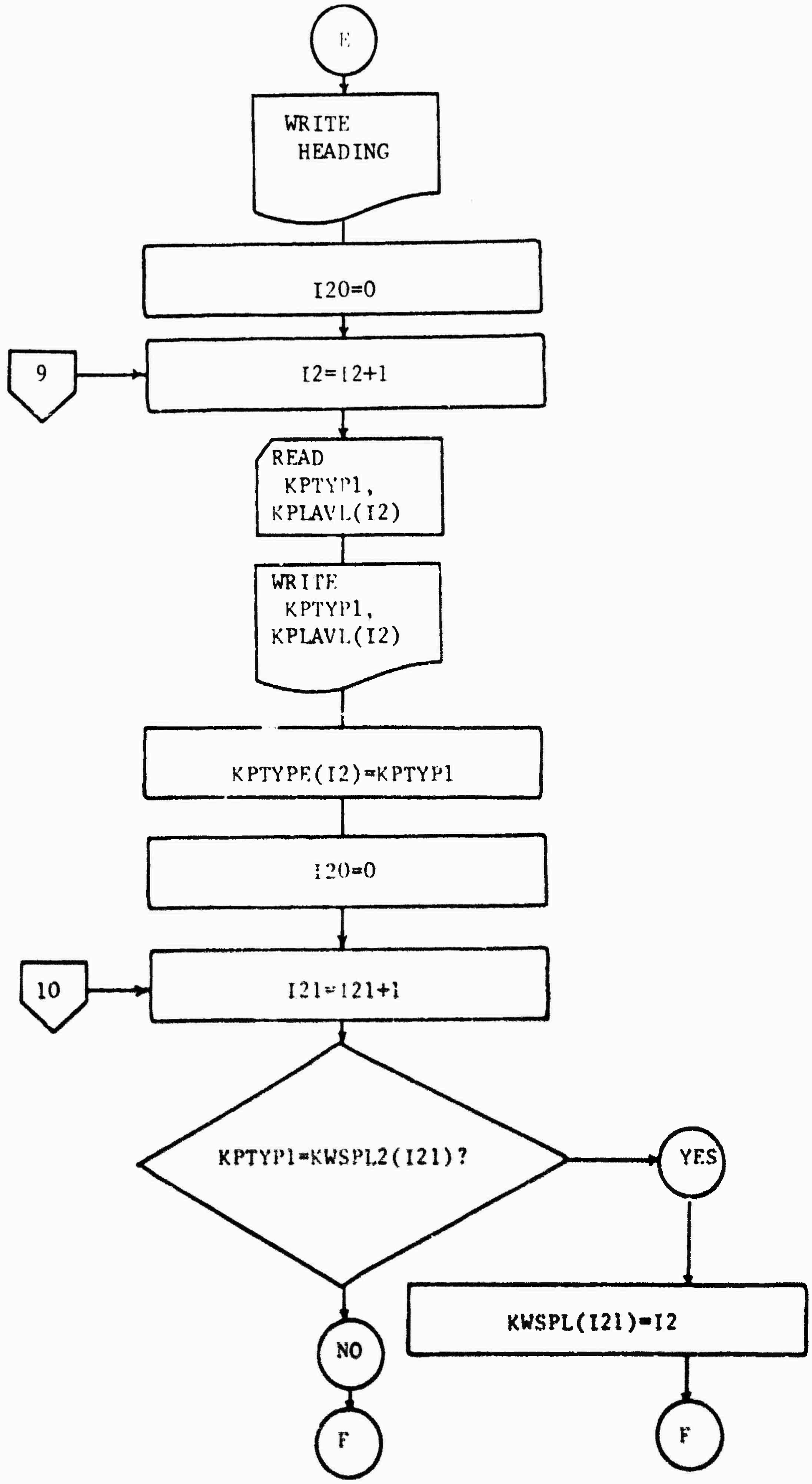

A
MODE1=MODE
MODE=3?
NO
YES
MODE1=1
CALL ORDER (MODE1,
J1HOLD,NASGN,C,E,IC)
CALL
OUTPUT
MODE=1?
YES
4
NO
2
MODE1=MODE
B

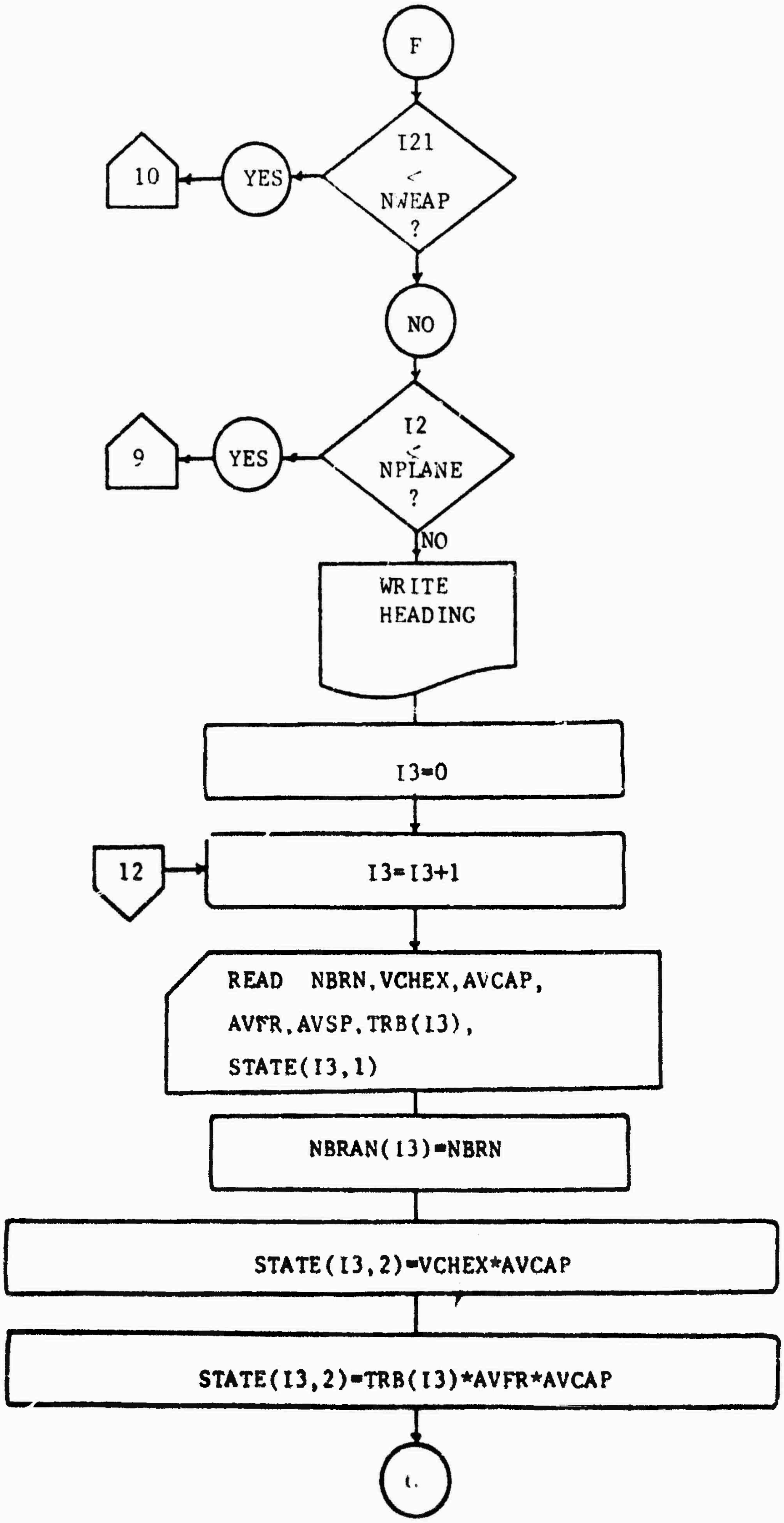


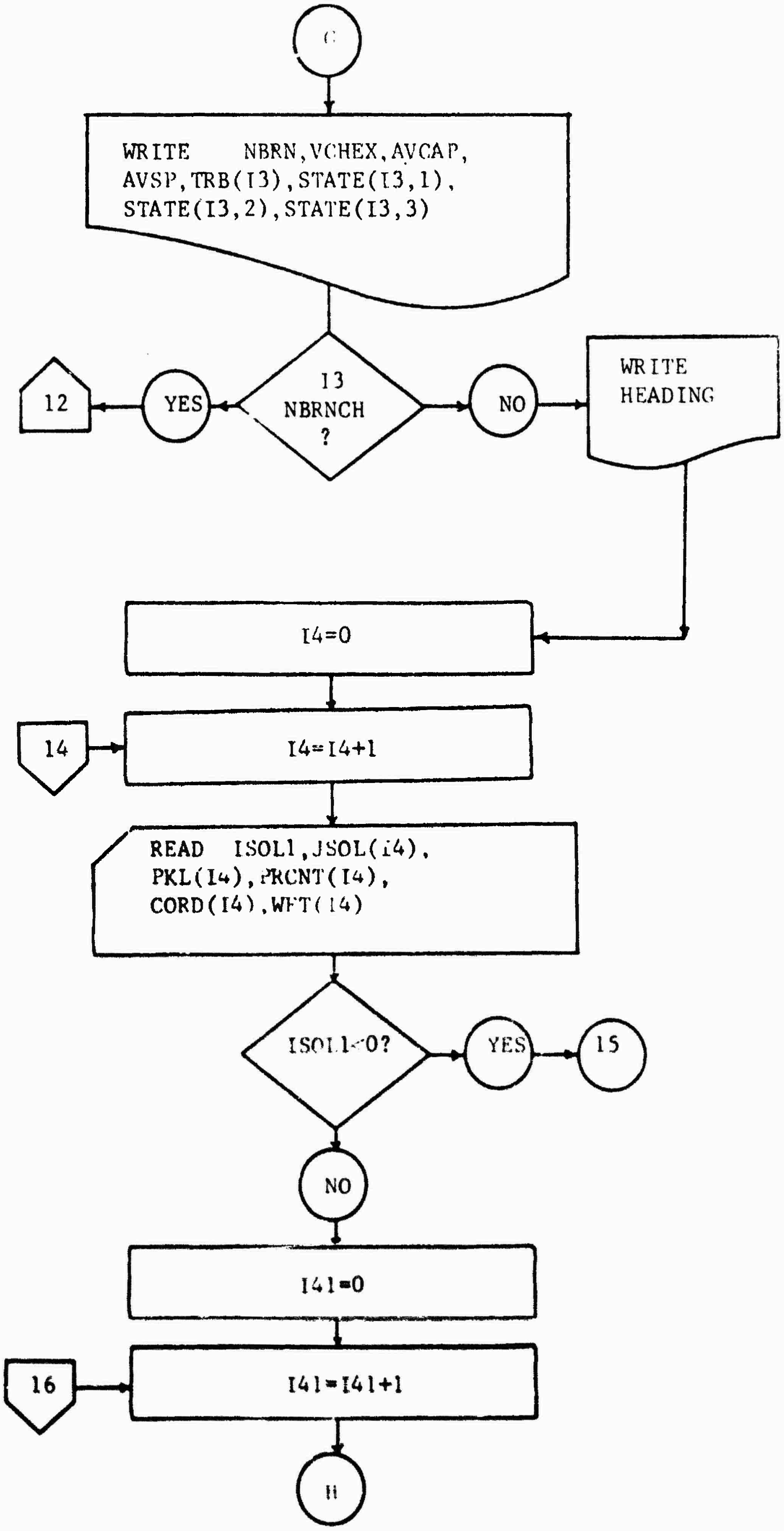
B
MODE=3?
NO
YES
MODE=2
CALL ORDER (MODE1, J1HOLD,NASGN,C,E,IC)
CALL OUTPUT
4
NASGN=NASGN+1
NASGN KASGN
YES
1
NO
END

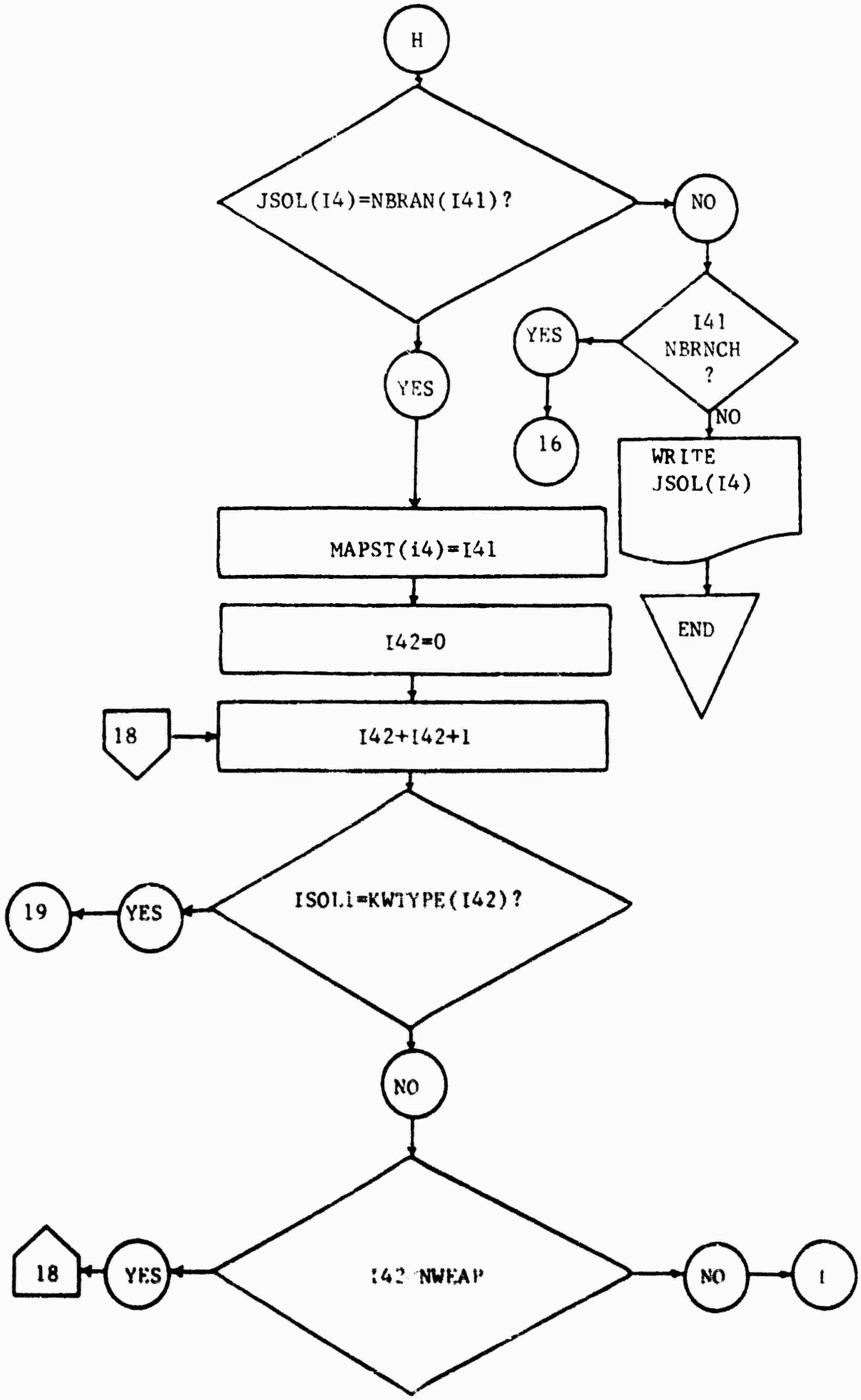

READIN
START
COMMON SAME AS MAIN
READ NASGN, KASGN
WRITE NASGN, KASGN
READ MODE, KSTOP, TOLE, TOLC, KUT
WRITE MODE, KSTOP, TOLE, TOLC, KUT
READ NWEAP,NPL-ANE,NBRNCH
WRITE NWEAP,NPLA-NE,NBRANCH
WRITE HEADING
D

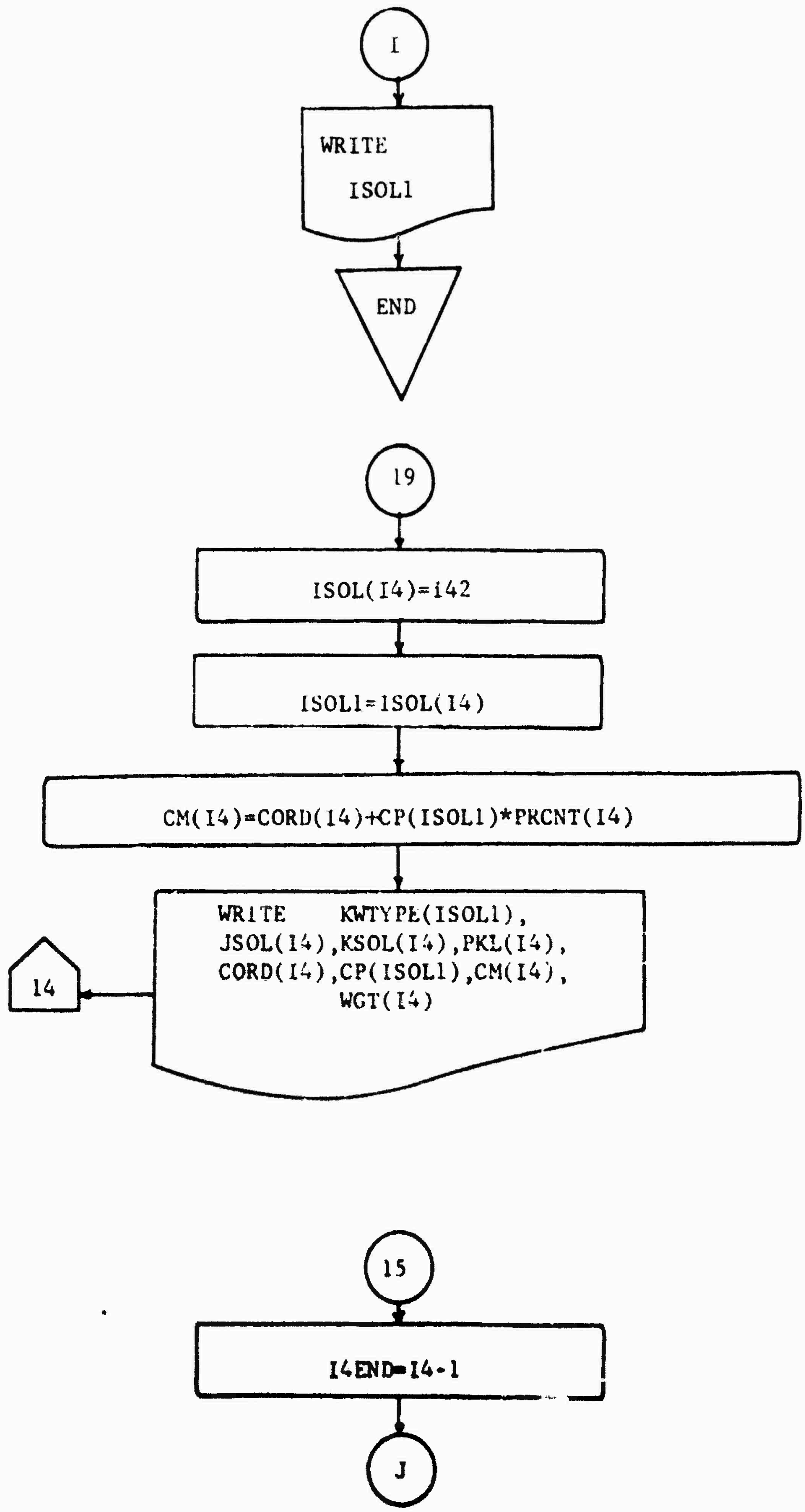

D
I1=1
6
I1=I1+1
READ KWTY-P1, KWSAVL(-I1), KWSPL(-I1), CP(I1)
WRITE KWTY-P1, KWSAVL(-I1), KWSPL(-I1), CP(I1)
6
YES
I1<NWEAP?
NO
WRITE HEADING
I20=0
7
I20=I20+1
KWSPL2(I20)=KWSPL(I20)
7
YES
I20<NWEAP?
NO
E


E
WRITE HEADING
I20=0
9
I2=I2+1
READ KPTYP1, KPLAVL(I2)
WRITE KPTYP1, KPLAVL(I2)
KPTYPE(I2)=KPTYP1
I20=0
10
I21=I21+1
KPTYP1=KWSPL2(I21)?
YES
KWSPL(I21)=I2
NO
F
F

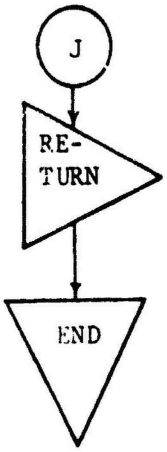

F
I21
<
NWEAP
?
YES
10
NO
I2
<
NPLANE
?
YES
9
NO
WRITE
HEADING
I3=0
12
I3=I3+1
READ NBRN,VCHEX,AVCAP,
AVFR,AVSP,TRB(I3),
STATE(I3,1)
NBRAN(I3)=NBRN
STATE(I3,2)=VCHEX*AVCAP
STATE(I3,2)=TRB(I3)*AVFR*AVCAP

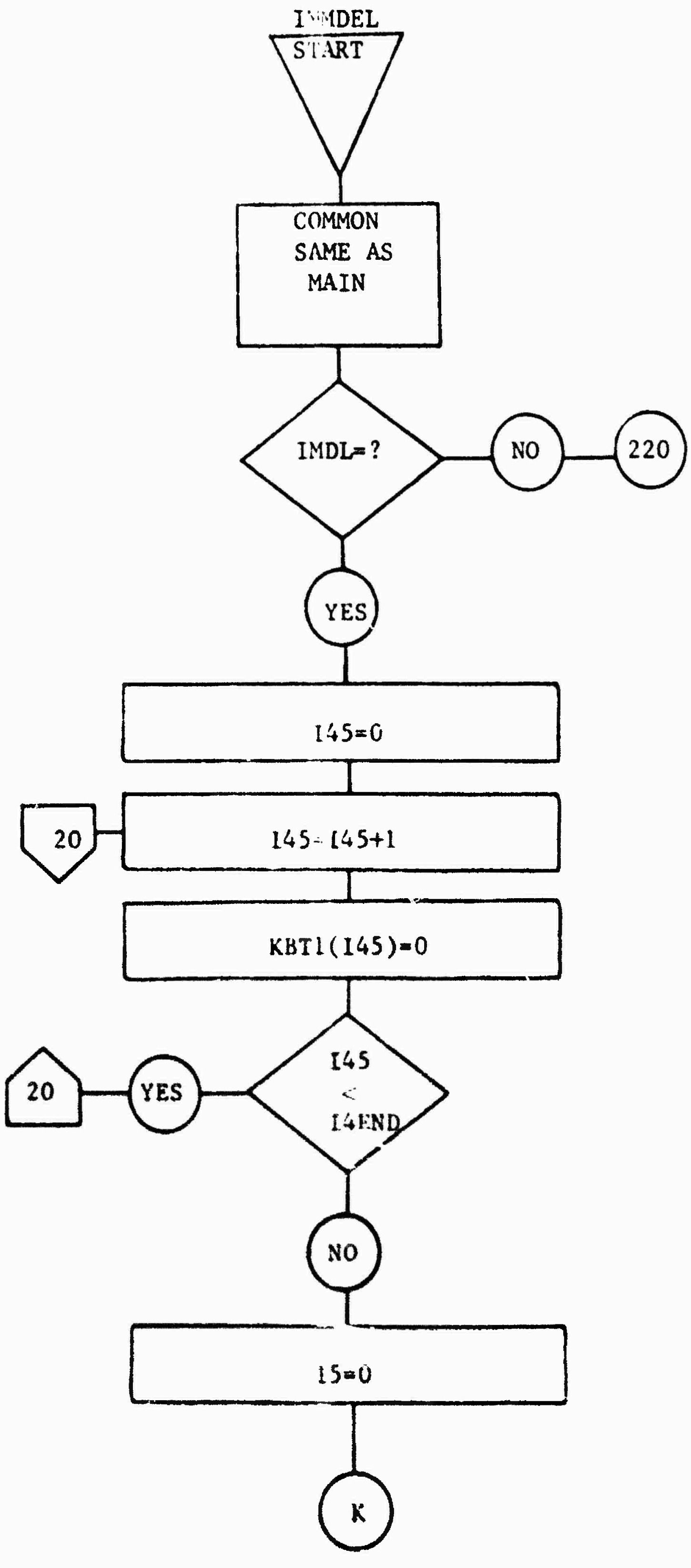

C
WRITE NBRN,VCHEX,AVCAP, AVSP,TRB(I3),STATE(I3,1), STATE(I3,2),STATE(I3,3)
12
YES
I3 NBRNCH ?
NO
WRITE HEADING
I4=0
14
I4=I4+1
READ ISOL1,JSOL(I4), PKL(I4),PRCNT(I4), CORD(I4),WFT(I4)
YES
15
NO
I41=0
16
I41=I41+1
H

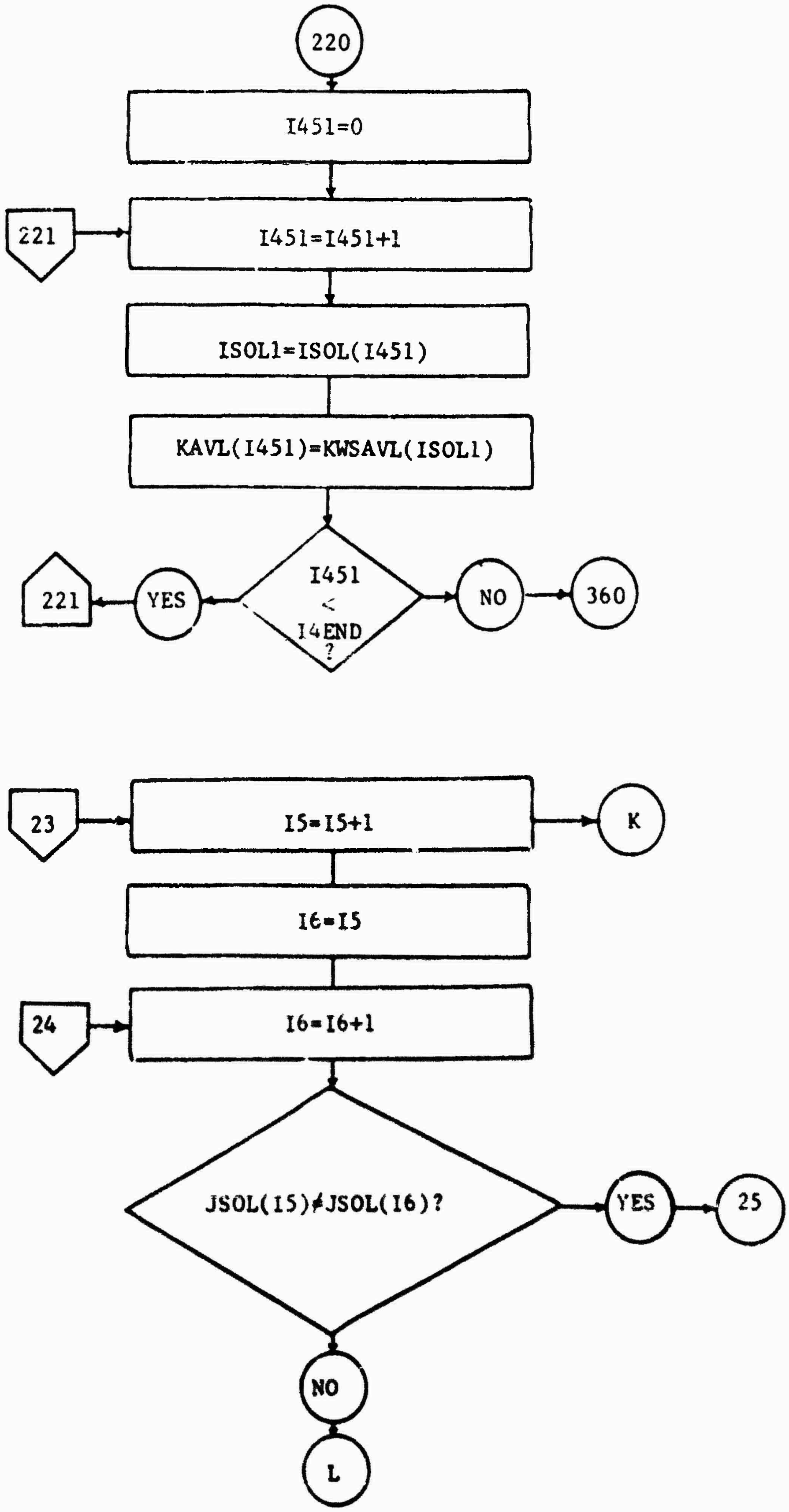

H
JSOL(I4)=NBRAN(I41)?
NO
I41
NBRNCH
?
YES
16
NO
WRITE
JSOL(I4)
END
YES
MAPST(i4)=I41
I42=0
18
I42+I42+1
ISOL1=KWTYPE(I42)?
YES
19
NO
I42 NWEAP
YES
18
NO
I

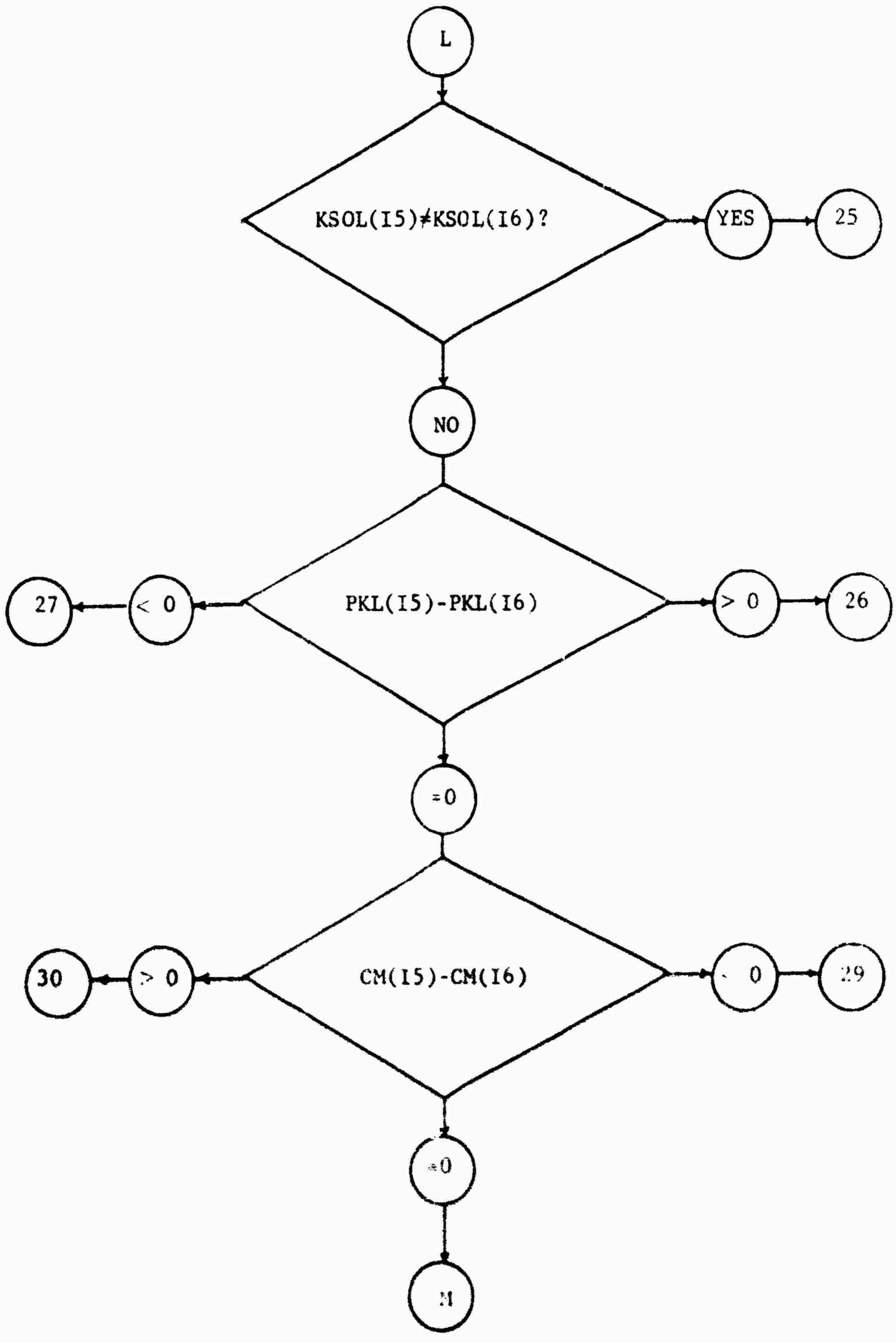


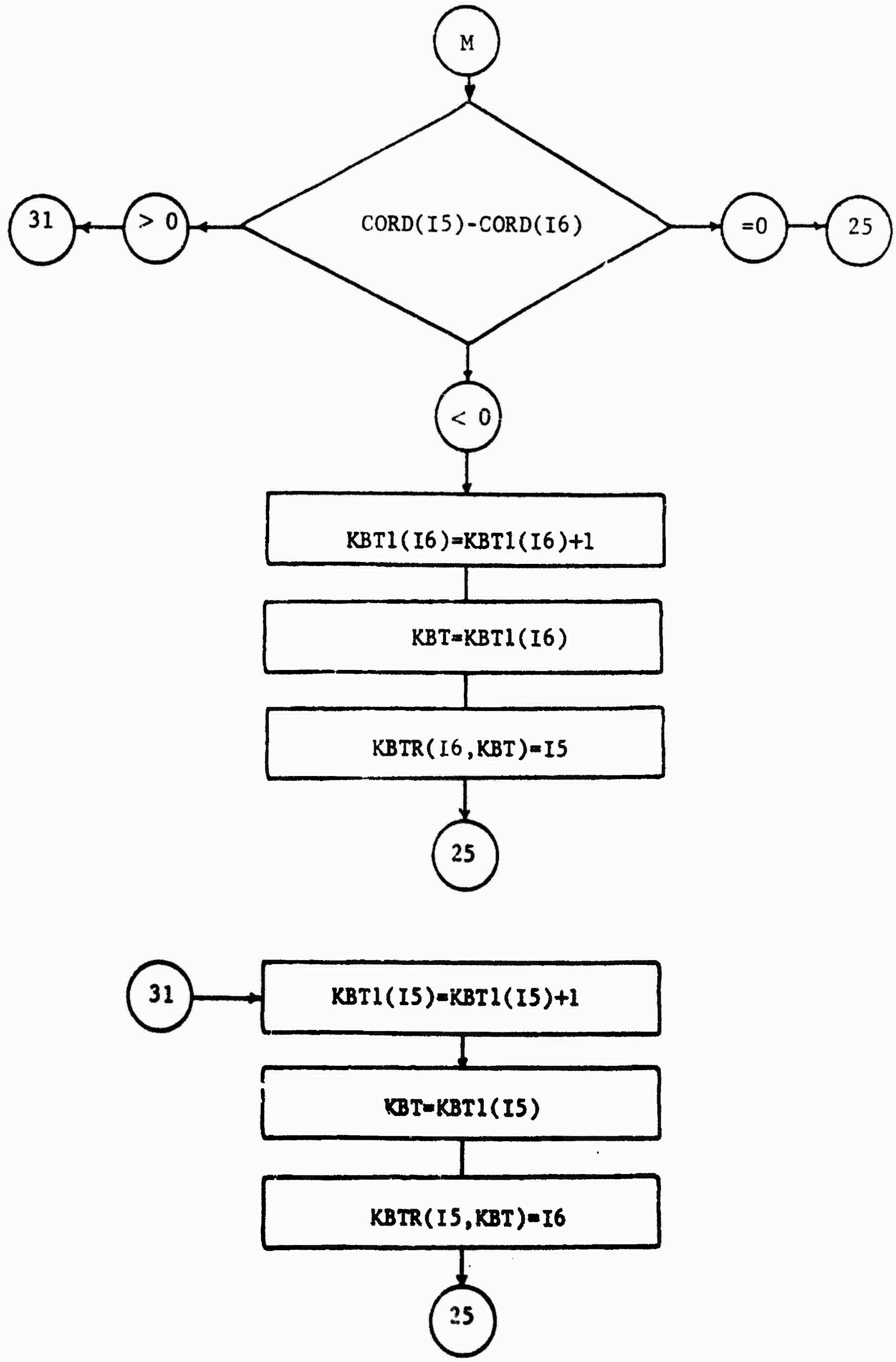
I
WRITE
ISOL1
END
19
ISOL(I4)=I42
ISOL1=ISOL(I4)
CM(I4)=CORD(I4)+CP(ISOL1)*PRCNT(I4)
WRITE KWTYPE(ISOL1), JSOL(I4),KSOL(I4),PKL(I4), CORD(I4),CP(ISOL1),CM(I4), WGT(I4)
14
15
I4END=I4-1
J

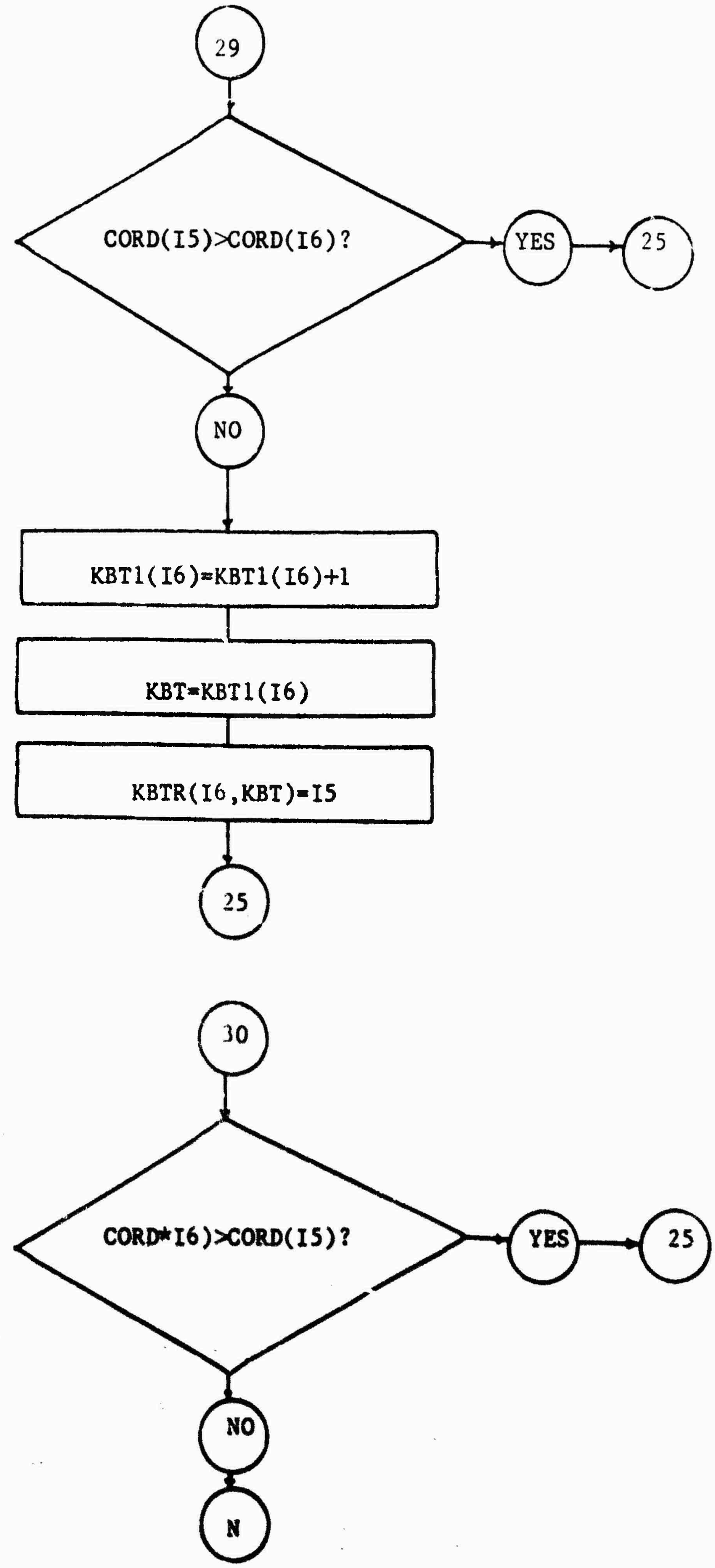

J
RE-
TURN
END


IMMDEL
START
COMMON
SAME AS
MAIN
IMDL=?
NO
220
YES
I45=0
20
I45=I45+1
KBT1(I45)=0
I45
<
I4END
YES
20
NO
I5=0
K

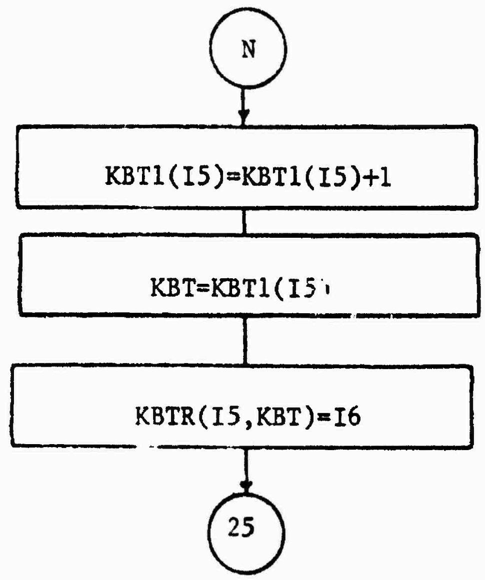

220
I451=0
221
I451=I451+1
ISOL1=ISOL(I451)
KAVL(I451)=KWSAVL(ISOL1)
I451 < I4END ?
YES
221
NO
360
23
I5=I5+1
K
I6=I5
24
I6=I6+1
JSOL(I5)≠JSOL(I6)?
YES
25
NO
L

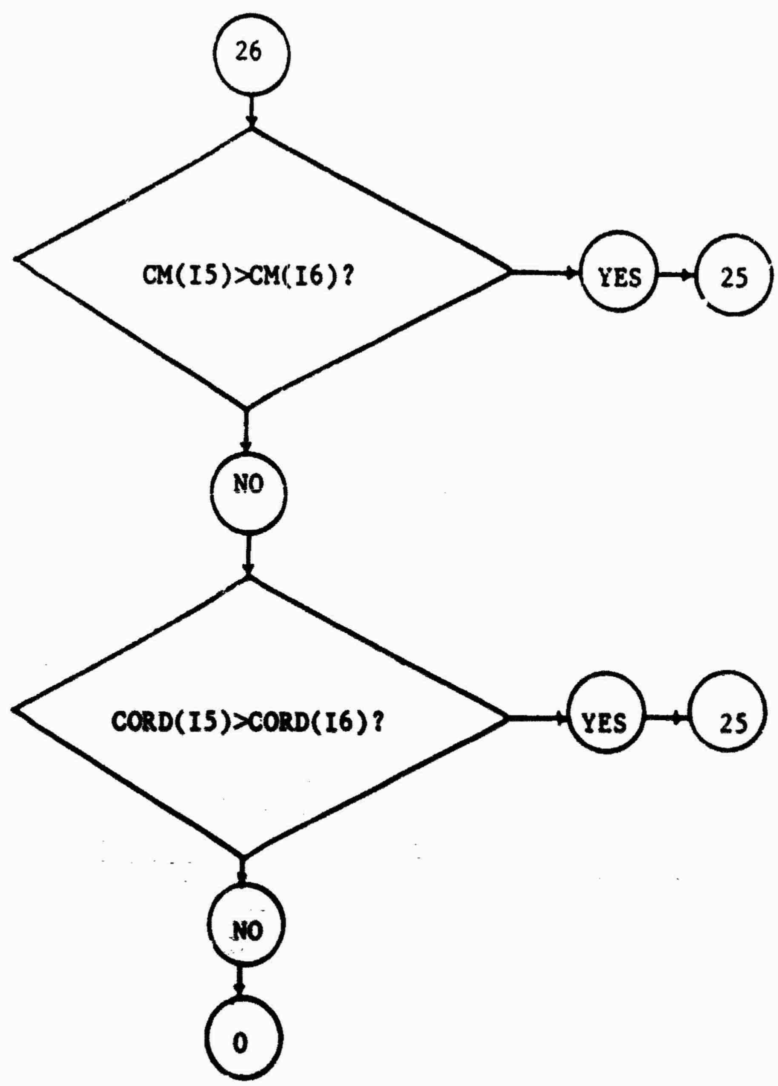

L
KSOL(I5)≠KSOL(I6)?
YES
25
NO
PKL(I5)-PKL(I6)
< 0
27
> 0
26
=0
CM(I5)-CM(I6)
> 0
30
0
29
=0
M

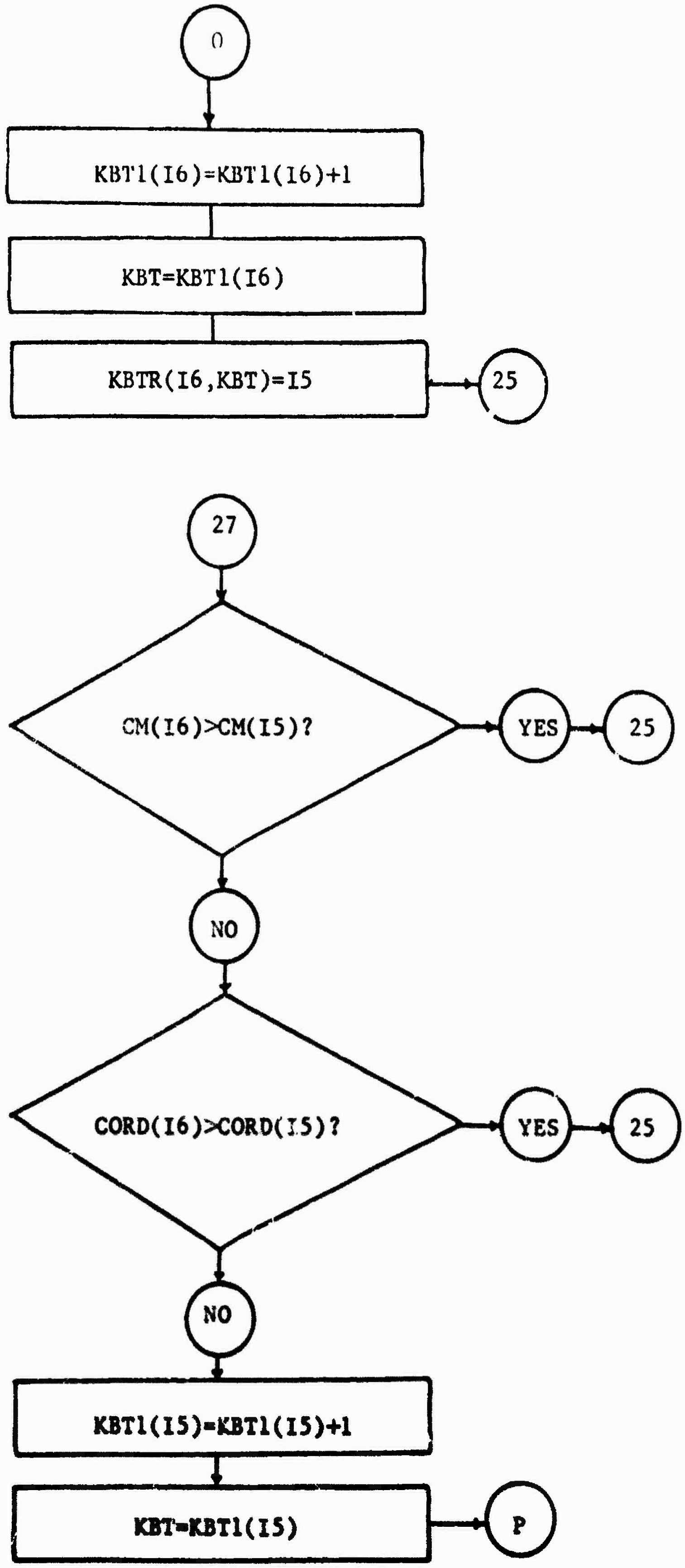

M
CORD(I5)-CORD(I6)
31
> 0
=0
25
< 0
KBT1(I6)=KBT1(I6)+1
KBT=KBT1(I6)
KBTR(I6,KBT)=I5
25
31
KBT1(I5)=KBT1(I5)+1
KBT=KBT1(I5)
KBTR(I5,KBT)=I6
25

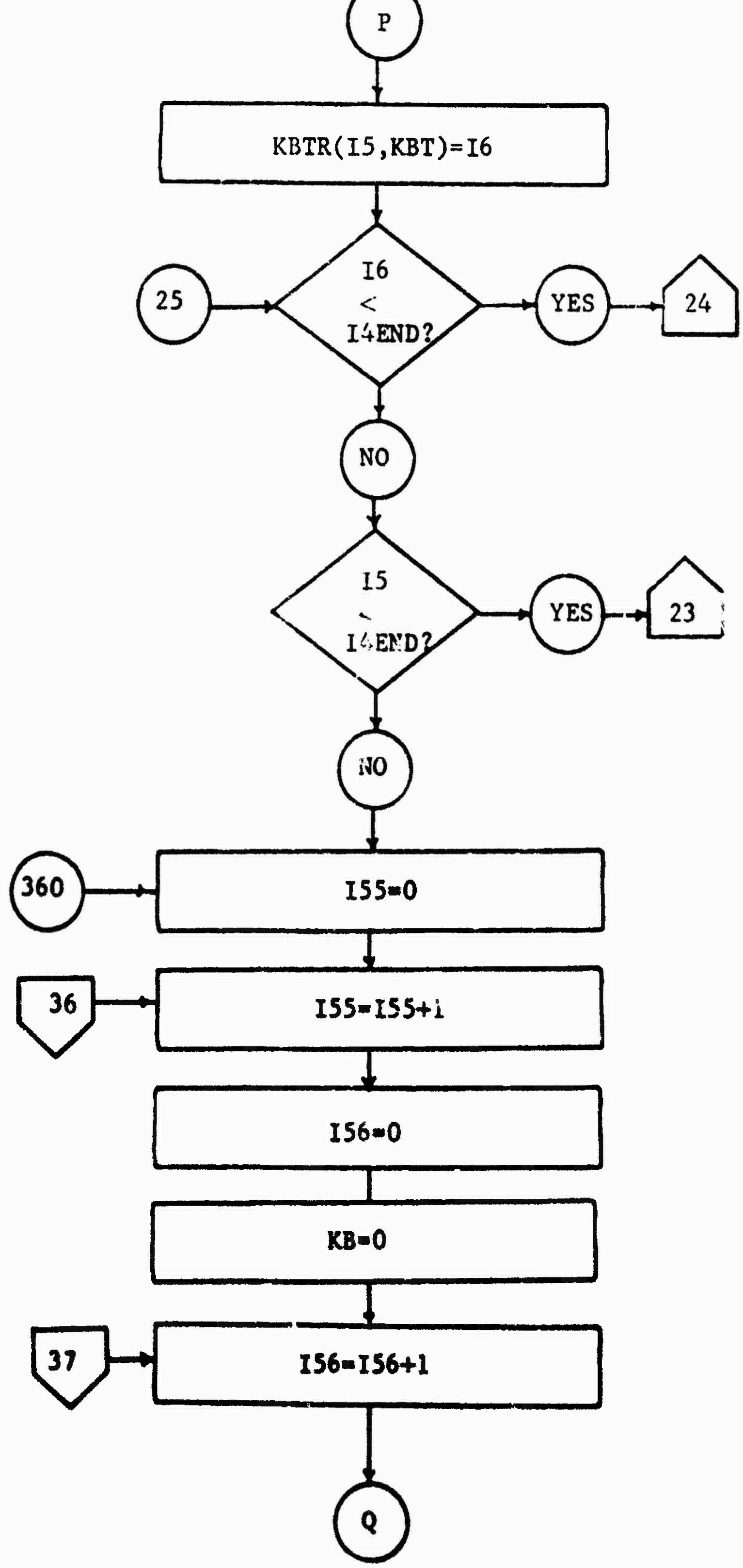


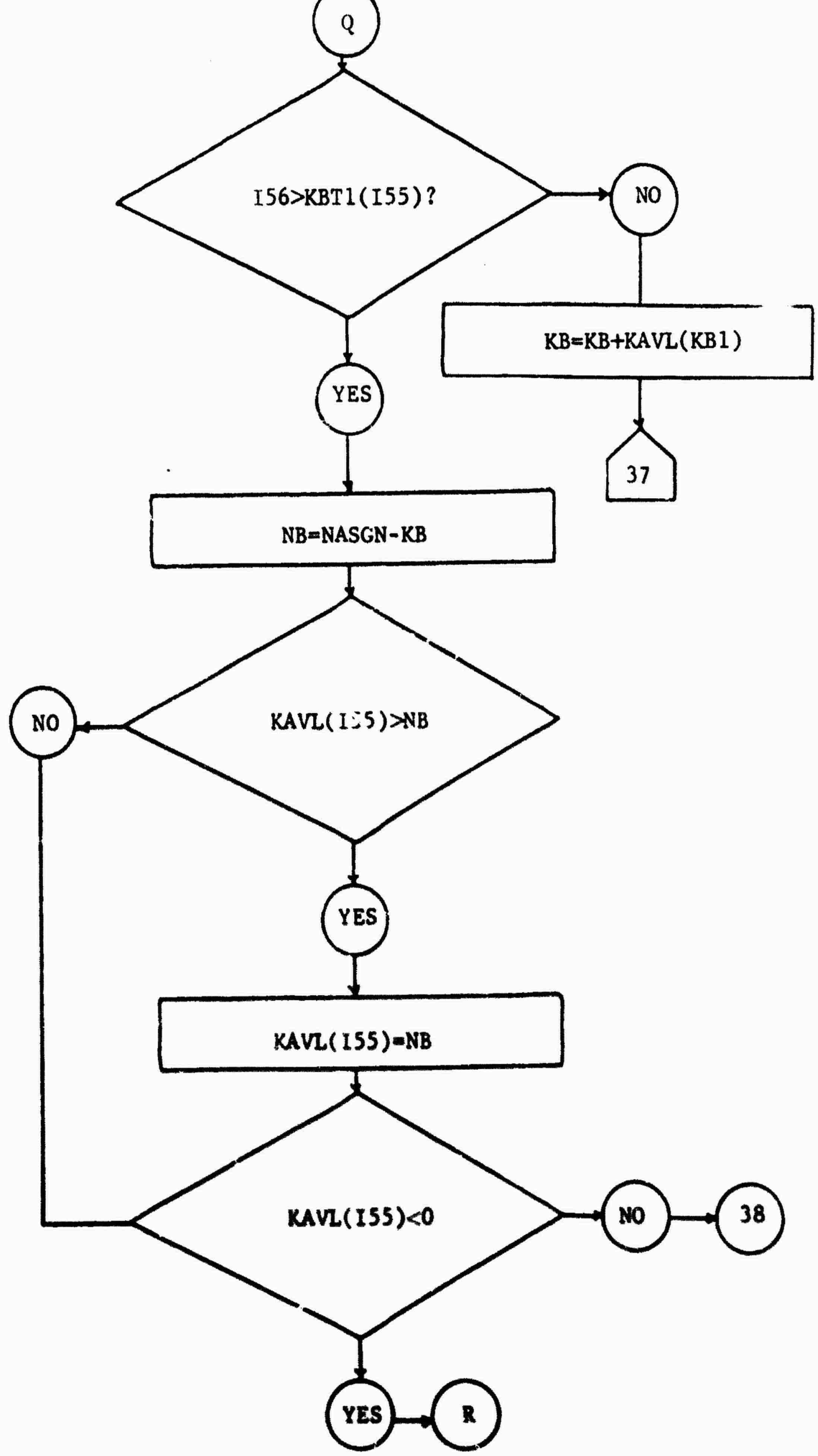
29
CORD(I5)>CORD(I6)?
YES
25
NO
KBT1(I6)=KBT1(I6)+1
KBT=KBT1(I6)
KBTR(I6,KBT)=I5
25
30
CORD*I6)>CORD(I5)?
YES
25
NO
N

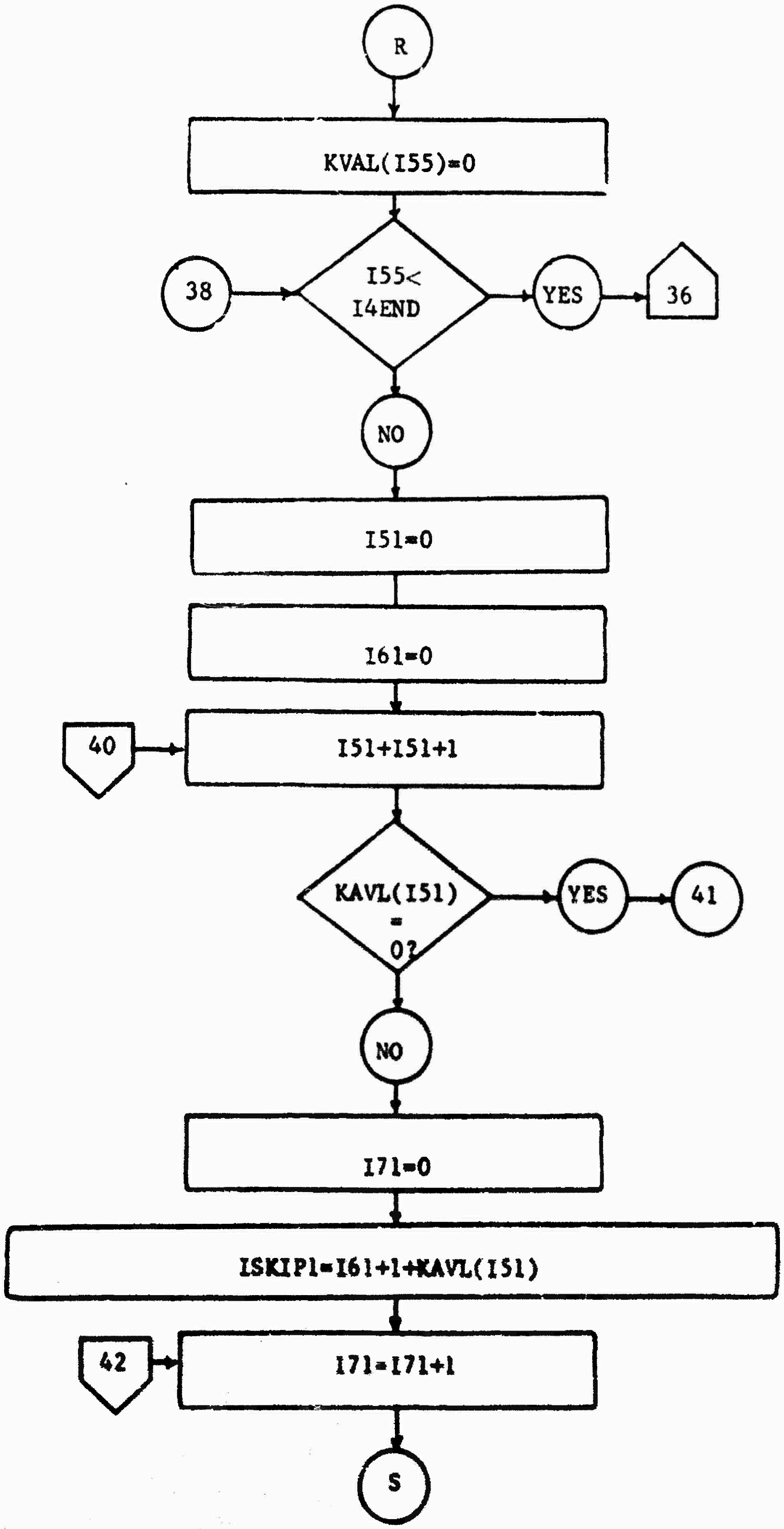

N
KBT1(I5)=KBT1(I5)+1
KBT=KBT1(I5)
KBTR(I5,KBT)=I6
25

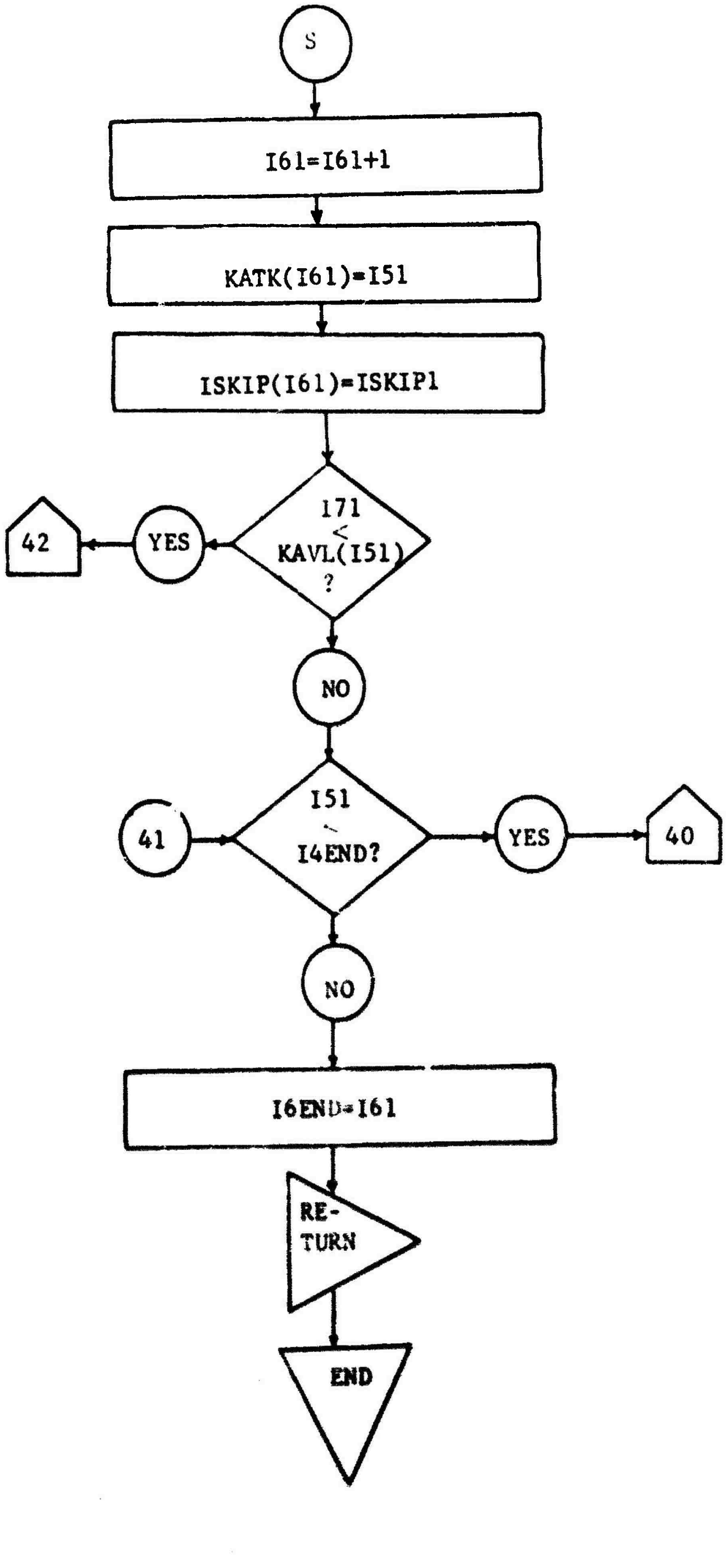

26
CM(I5)>CM(I6)?
YES
25
NO
CORD(I5)>CORD(I6)?
YES
25
NO
0

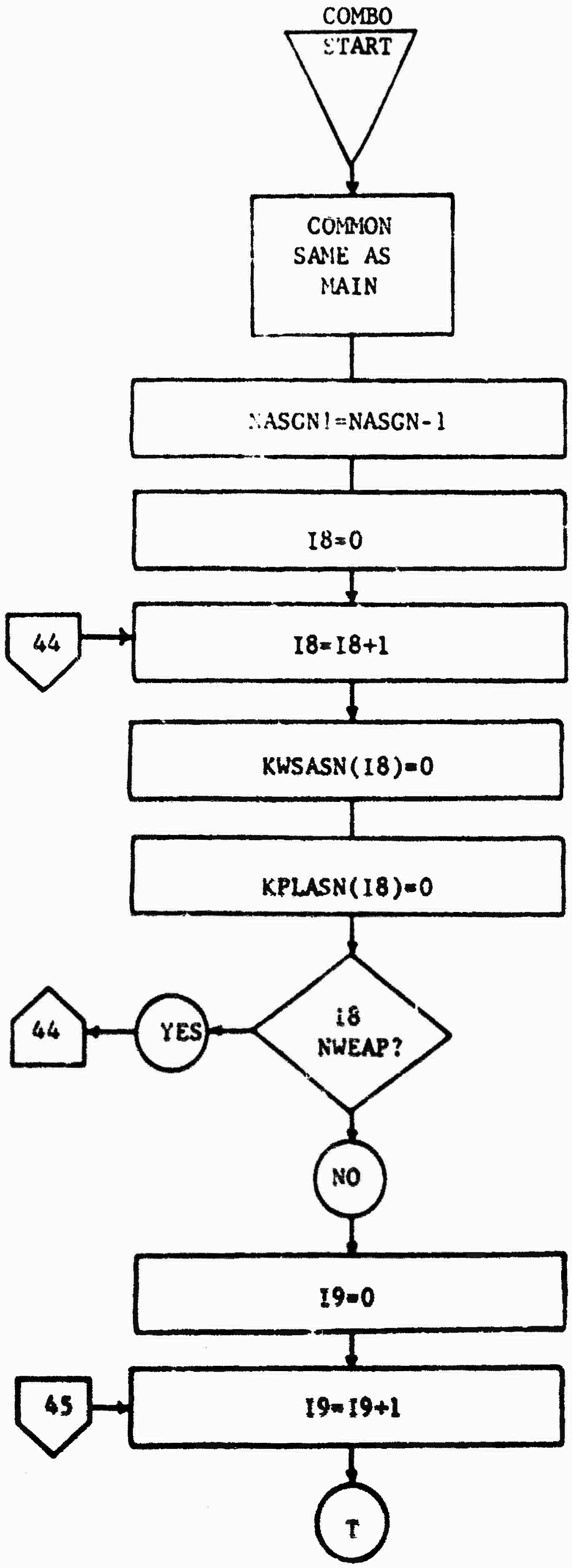

0
KBT1(I6)=KBT1(I6)+1
KBT=KBT1(I6)
KBTR(I6,KBT)=I5
25
27
CM(I6)>CM(I5)?
YES
25
NO
CORD(I6)>CORD(I5)?
YES
25
NO
KBT1(I5)=KBT1(I5)+1
KBT=KBT1(I5)
P

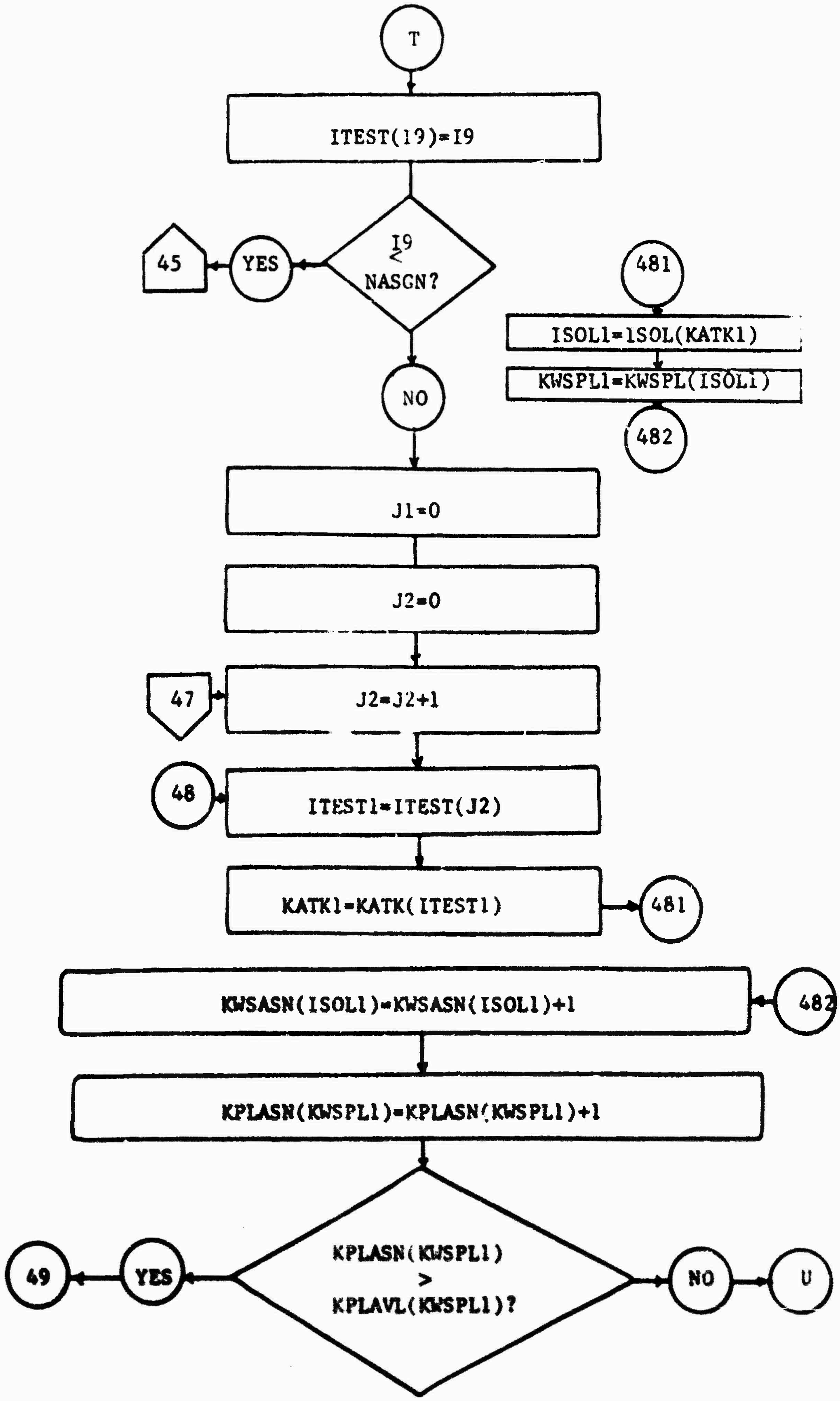

P
KBTR(I5,KBT)=I6
25
I6
<
I4END?
YES
24
NO
I5
<
I4END?
YES
23
NO
360
I55=0
36
I55=I55+1
I56=0
KB=0
37
I56=I56+1
Q

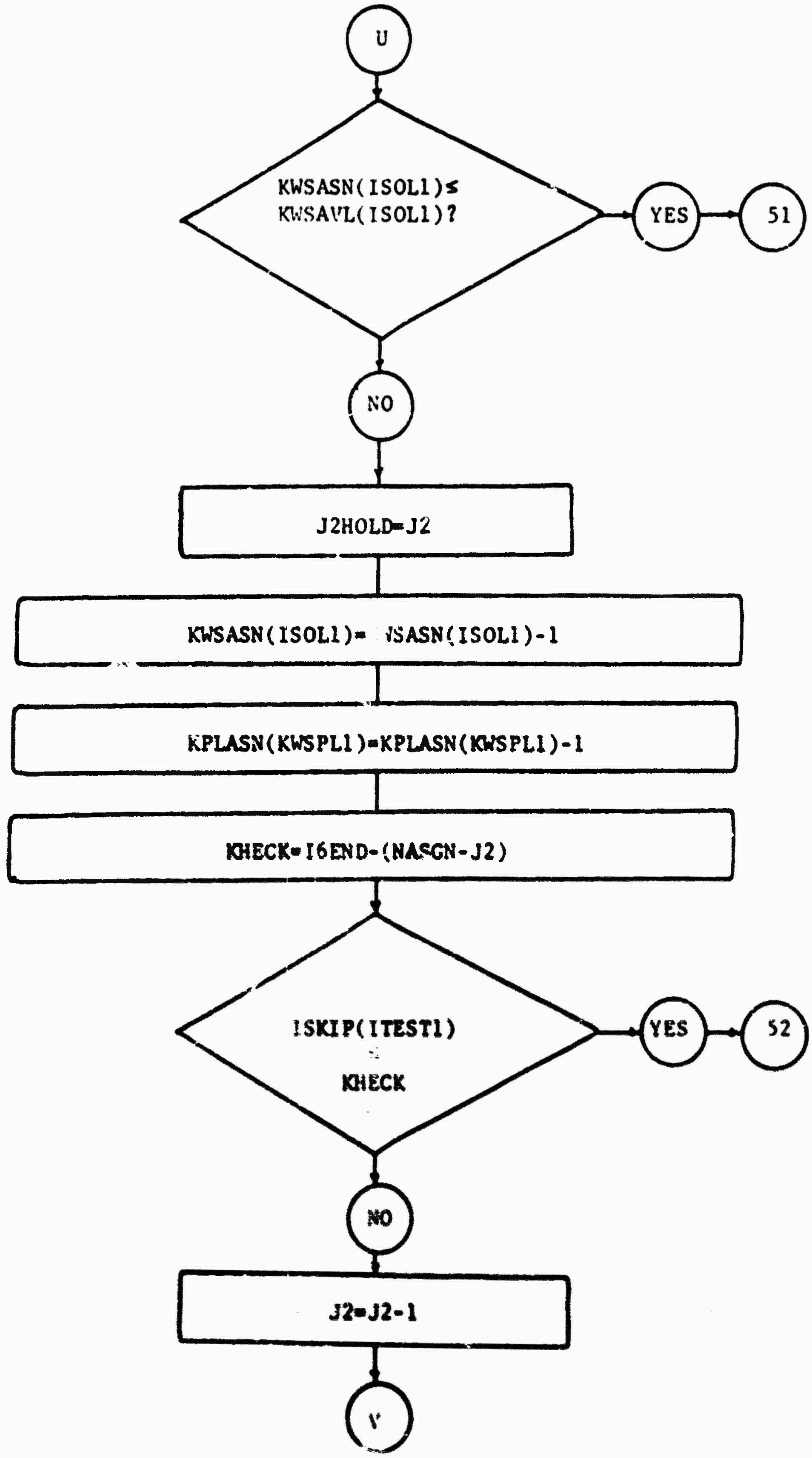

Q
I56>KBT1(I55)?
NO
KB=KB+KAVL(KB1)
37
YES
NB=NASGN-KB
KAVL(I55)>NB
NO
YES
KAVL(I55)=NB
KAVL(I55)<0
NO
38
YES
R

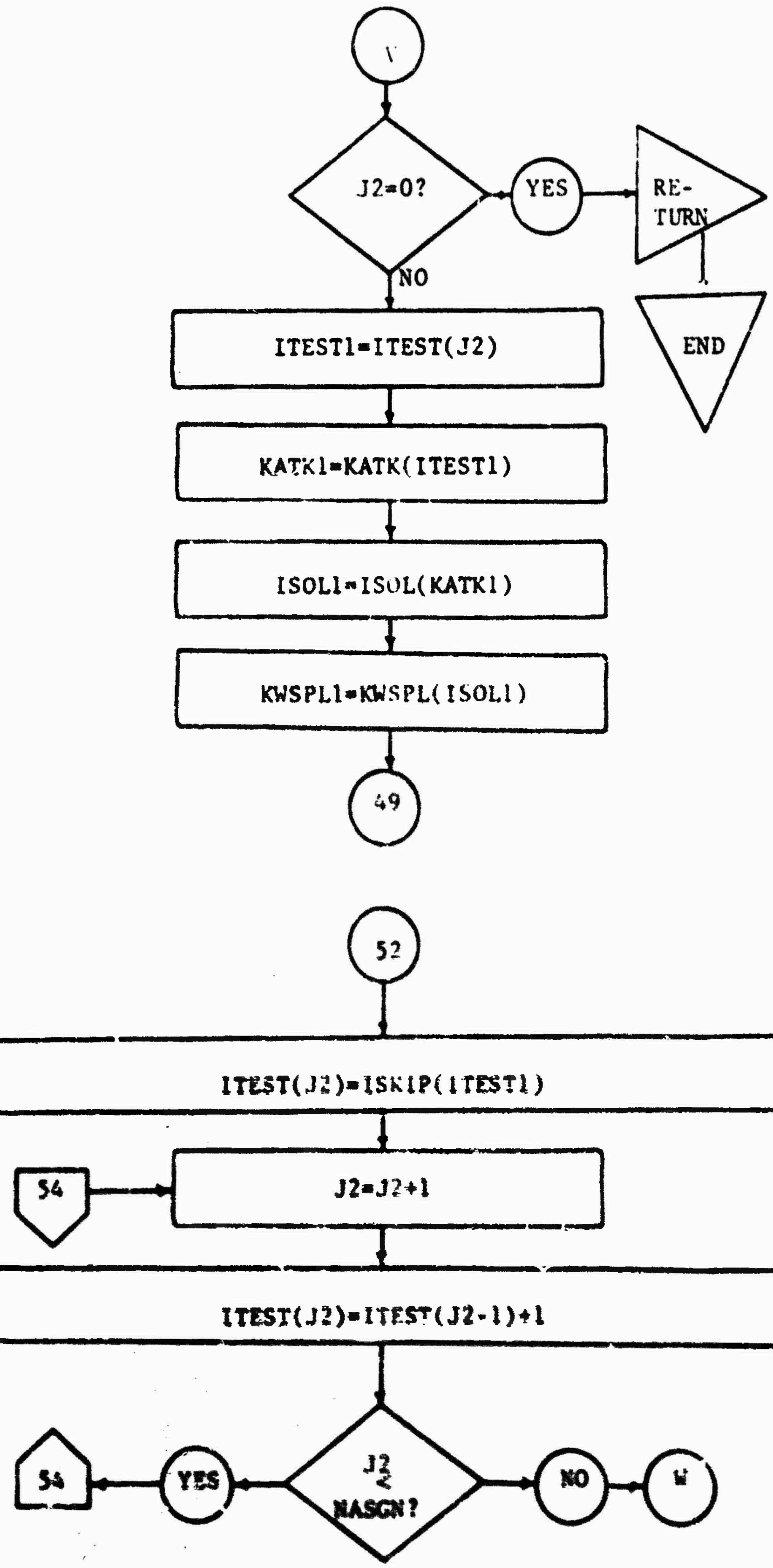

R
KVAL(I55)=0
38
I55<
I4END
YES
36
NO
I51=0
I61=0
40
I51+I51+1
KAVL(I51)
=
0?
YES
41
NO
I71=0
ISKIP1=I61+1+KAVL(I51)
42
I71=I71+1
S

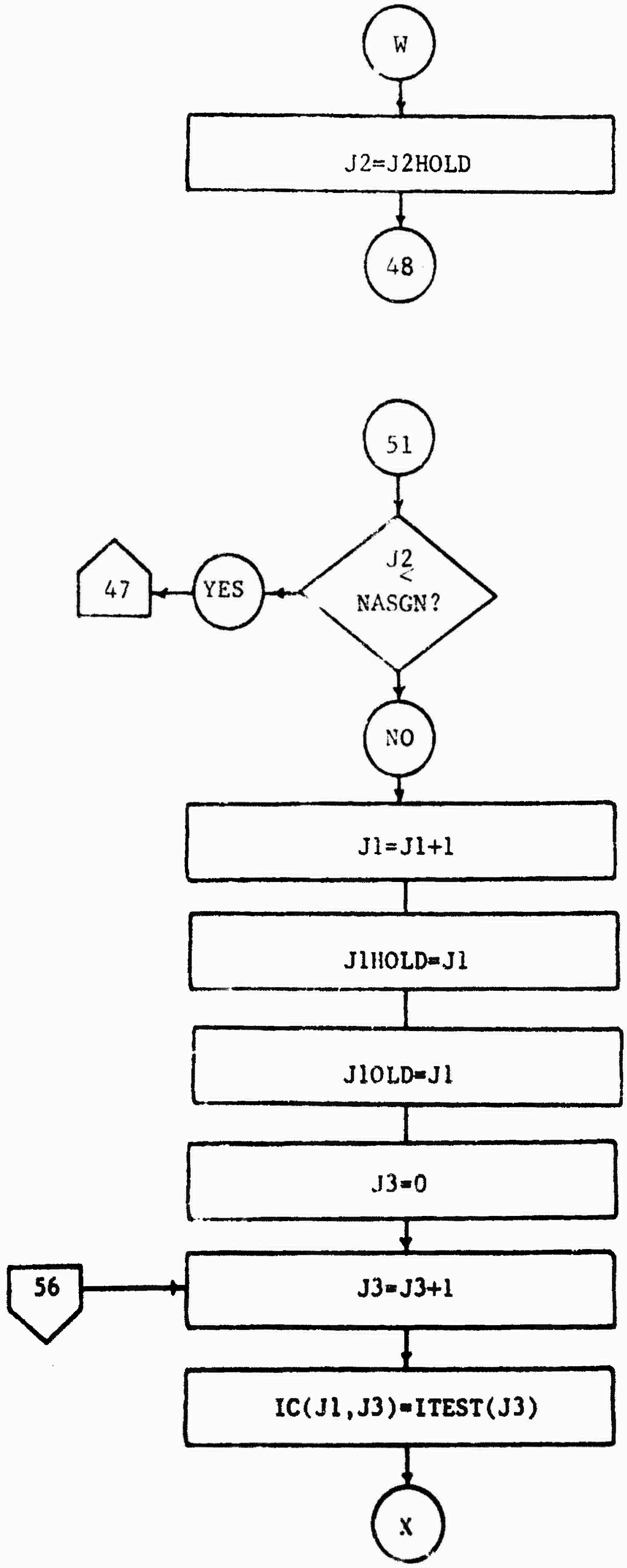

S
I61=I61+1
KATK(I61)=I51
ISKIP(I61)=ISKIP1
I71 < KAVL(I51) ?
YES
42
NO
I51 ~ I4END?
41
YES
40
NO
I6END=I61
RE-
TURN
END

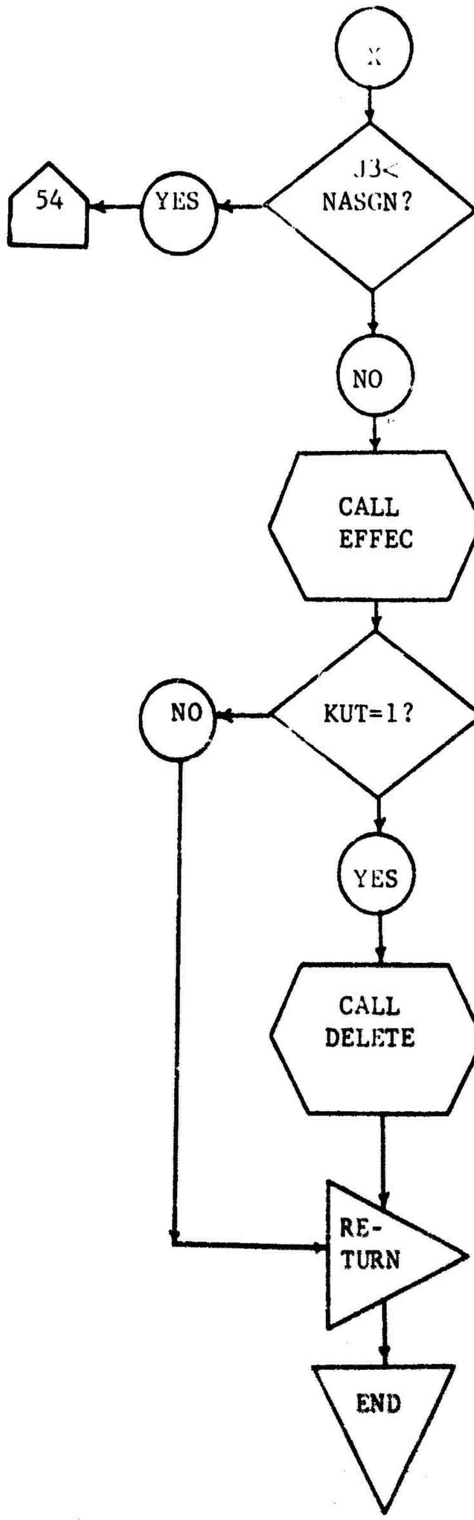

COMBO
START
COMMON
SAME AS
MAIN
NASCN1=NASCN-1
I8=0
44
I8=I8+1
KWSASN(I8)=0
KPLASN(I8)=0
44
YES
I8
NWEAP?
NO
I9=0
45
I9=I9+1
T

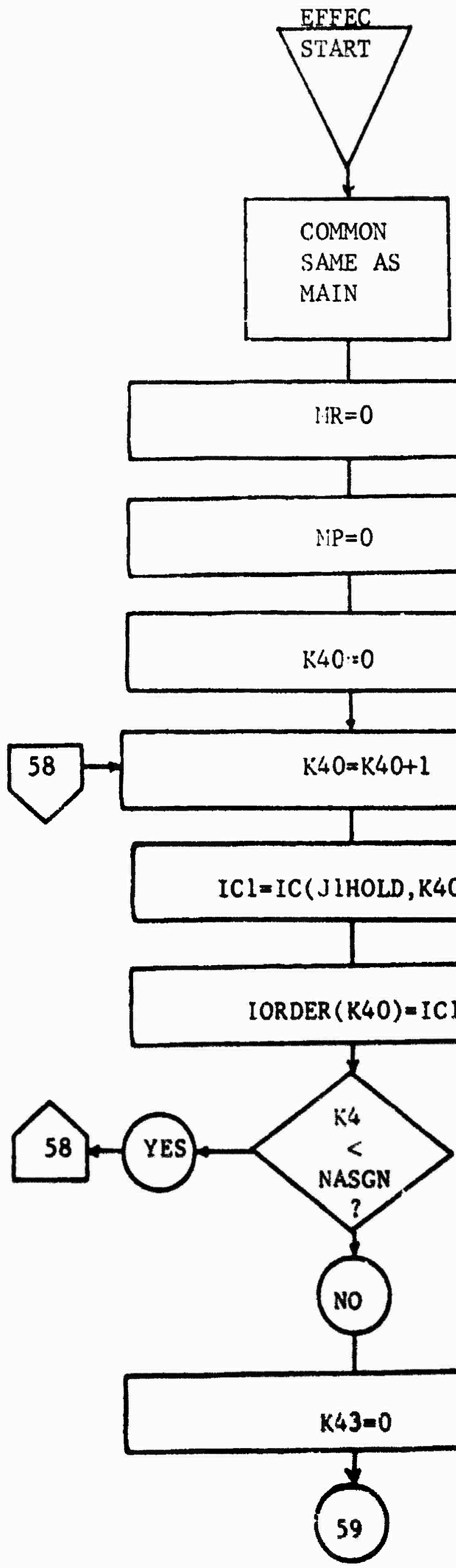

T
ITEST(19)=I9
I9 < NASGN?
YES
45
NO
481
ISOL1=ISOL(KATK1)
KWSPL1=KWSPL(ISOL1)
482
J1=0
J2=0
47
J2=J2+1
48
ITEST1=ITEST(J2)
KATK1=KATK(ITEST1)
481
KWSASN(ISOL1)=KWSASN(ISOL1)+1
482
KPLASN(KWSPL1)=KPLASN(KWSPL1)+1
KPLASN(KWSPL1) > KPLAVL(KWSPL1)?
YES
49
NO
U

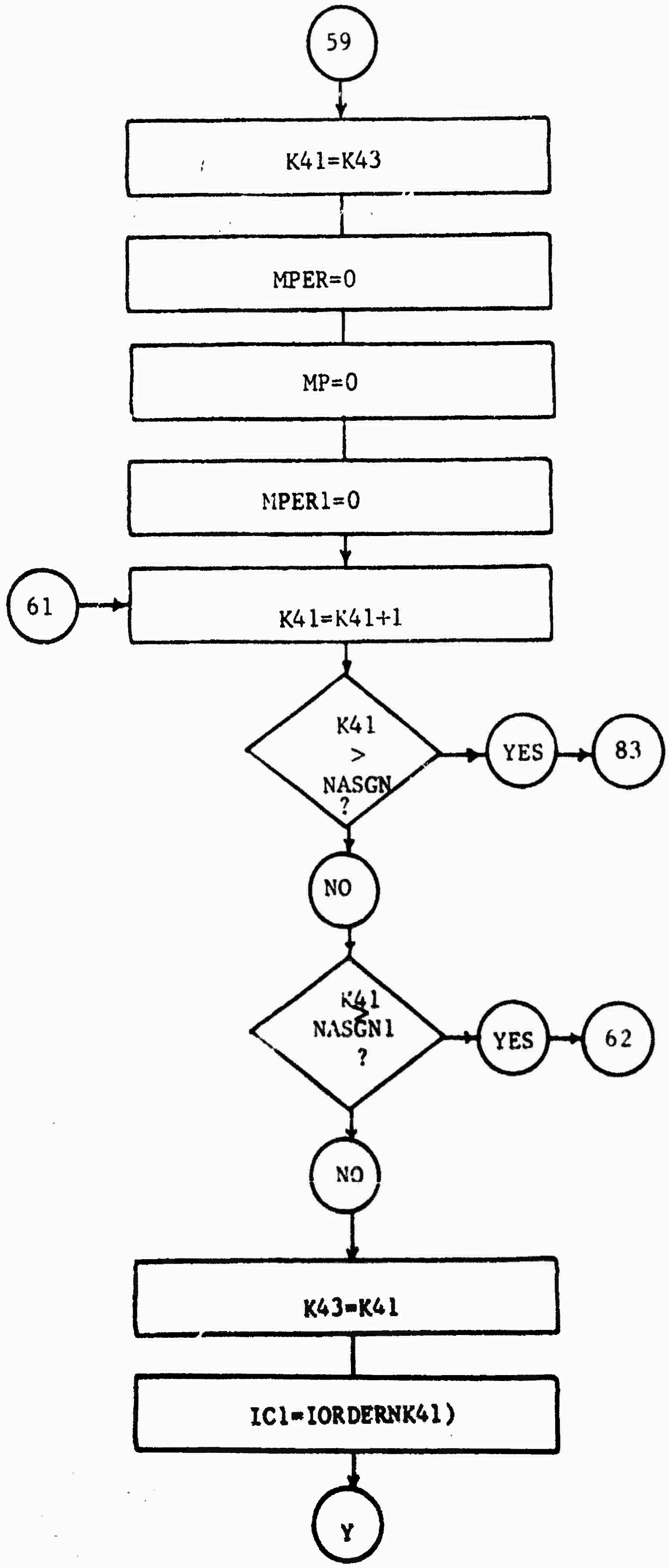

U
KWSASN(ISOL1)≤
KWSAVL(ISOL1)?
YES
51
NO
J2HOLD=J2
KWSASN(ISOL1)= WSASN(ISOL1)-1
KPLASN(KWSPL1)=KPLASN(KWSPL1)-1
KHECK=I6END-(NASGN-J2)
ISKIP(ITEST1)
KHECK
YES
52
NO
J2=J2-1
V

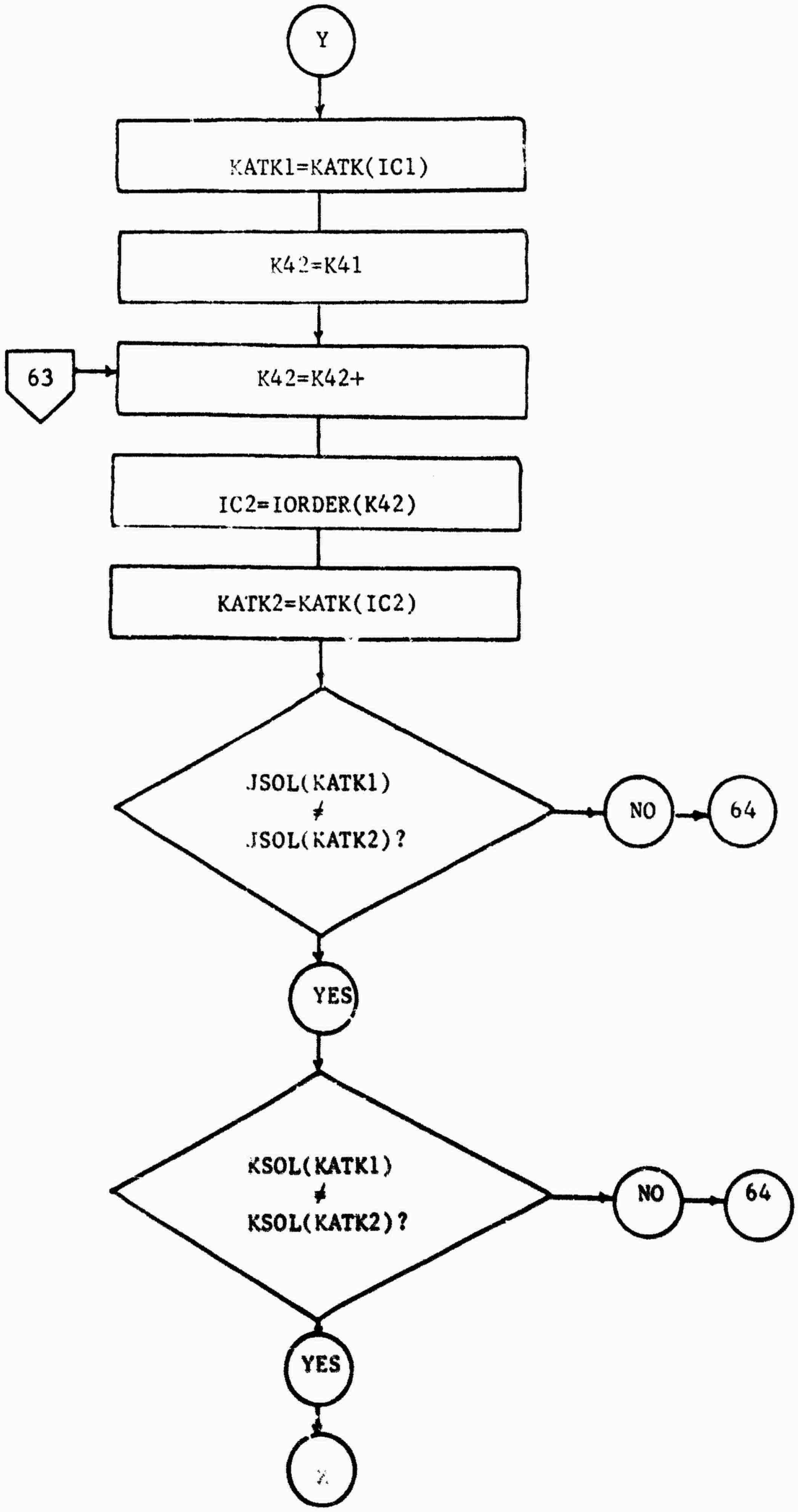

V
J2=0?
YES
RE-
TURN
NO
END
ITEST1=ITEST(J2)
KATK1=KATK(ITEST1)
ISOL1=ISOL(KATK1)
KWSPL1=KWSPL(ISOL1)
49
52
ITEST(J2)=ISKIP(ITEST1)
54
J2=J2+1
ITEST(J2)=ITEST(J2-1)+1
54
YES
J2
<
NASGN?
NO
W

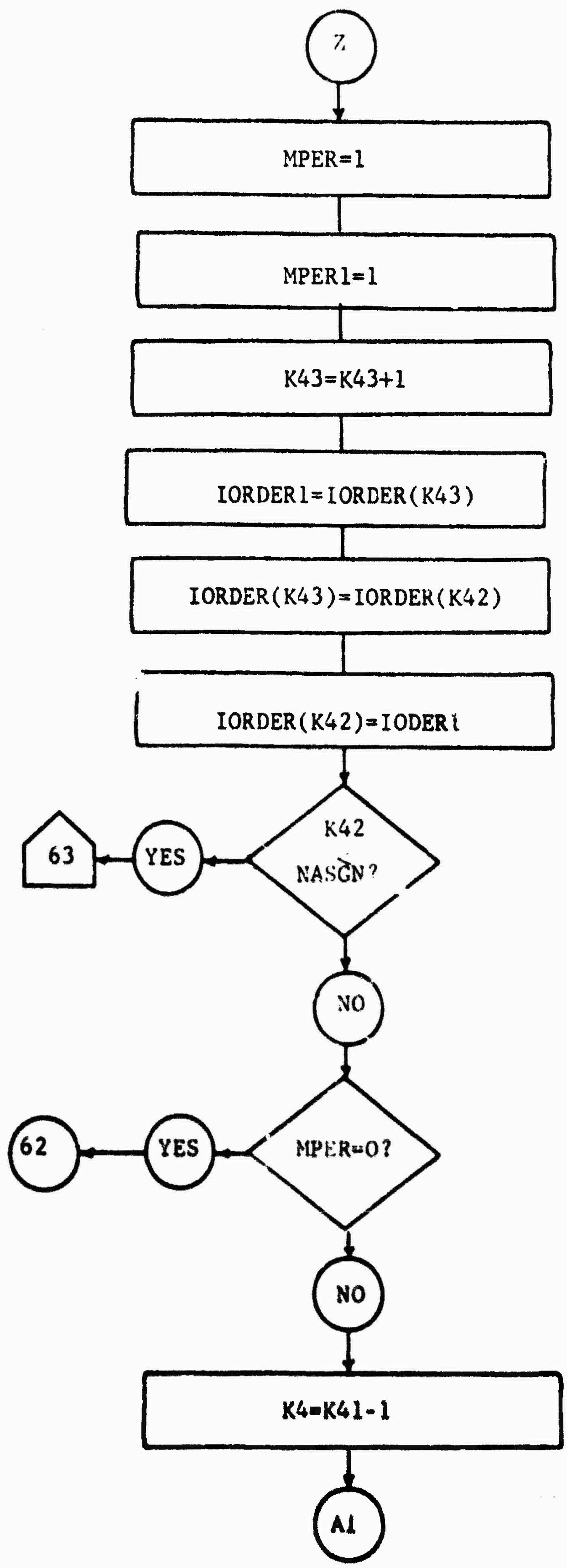

W
J2=J2HOLD
48
51
J2 ≤ NASGN?
YES
47
NO
J1=J1+1
J1HOLD=J1
J1OLD=J1
J3=0
56
J3=J3+1
IC(J1,J3)=ITEST(J3)
X

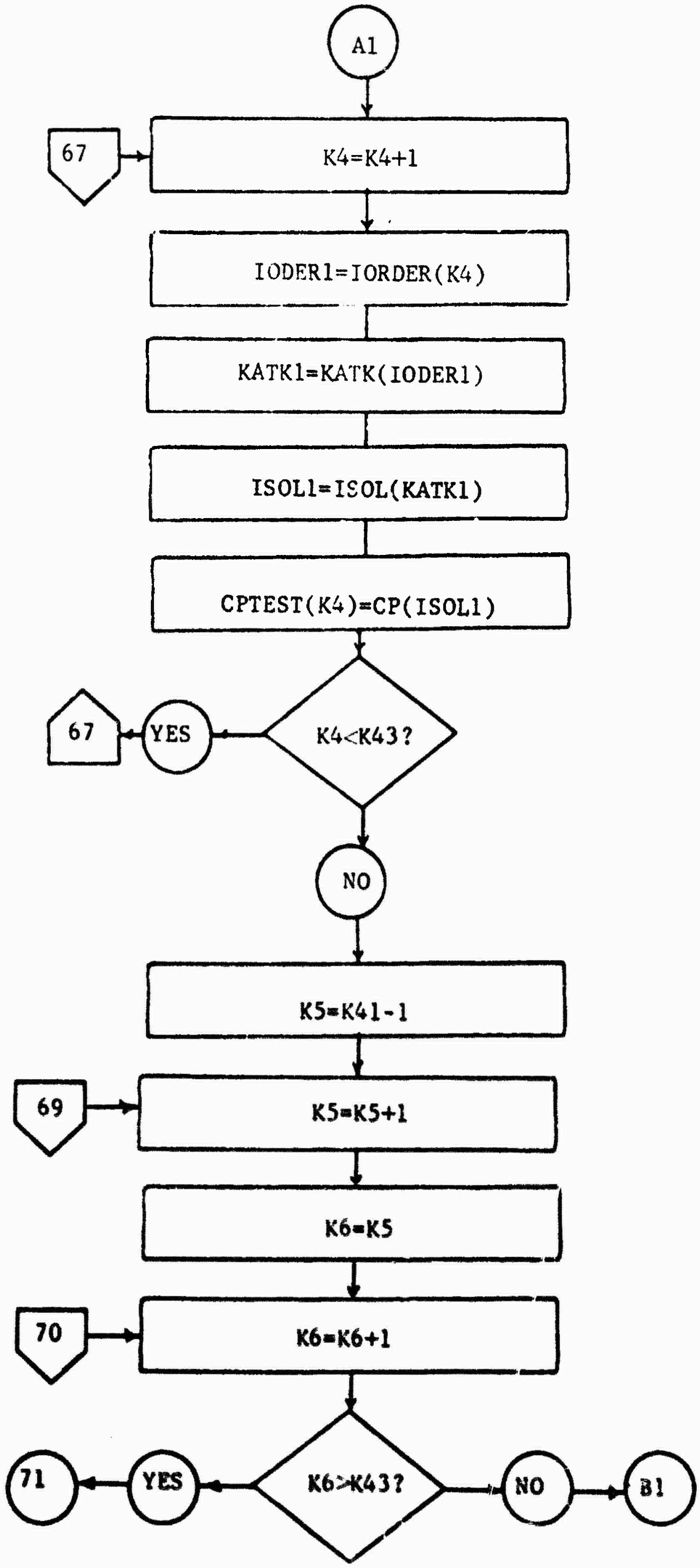

X
J3<
NASGN?
YES
54
NO
CALL
EFFEC
KUT=1?
NO
YES
CALL
DELETE
RE-
TURN
END

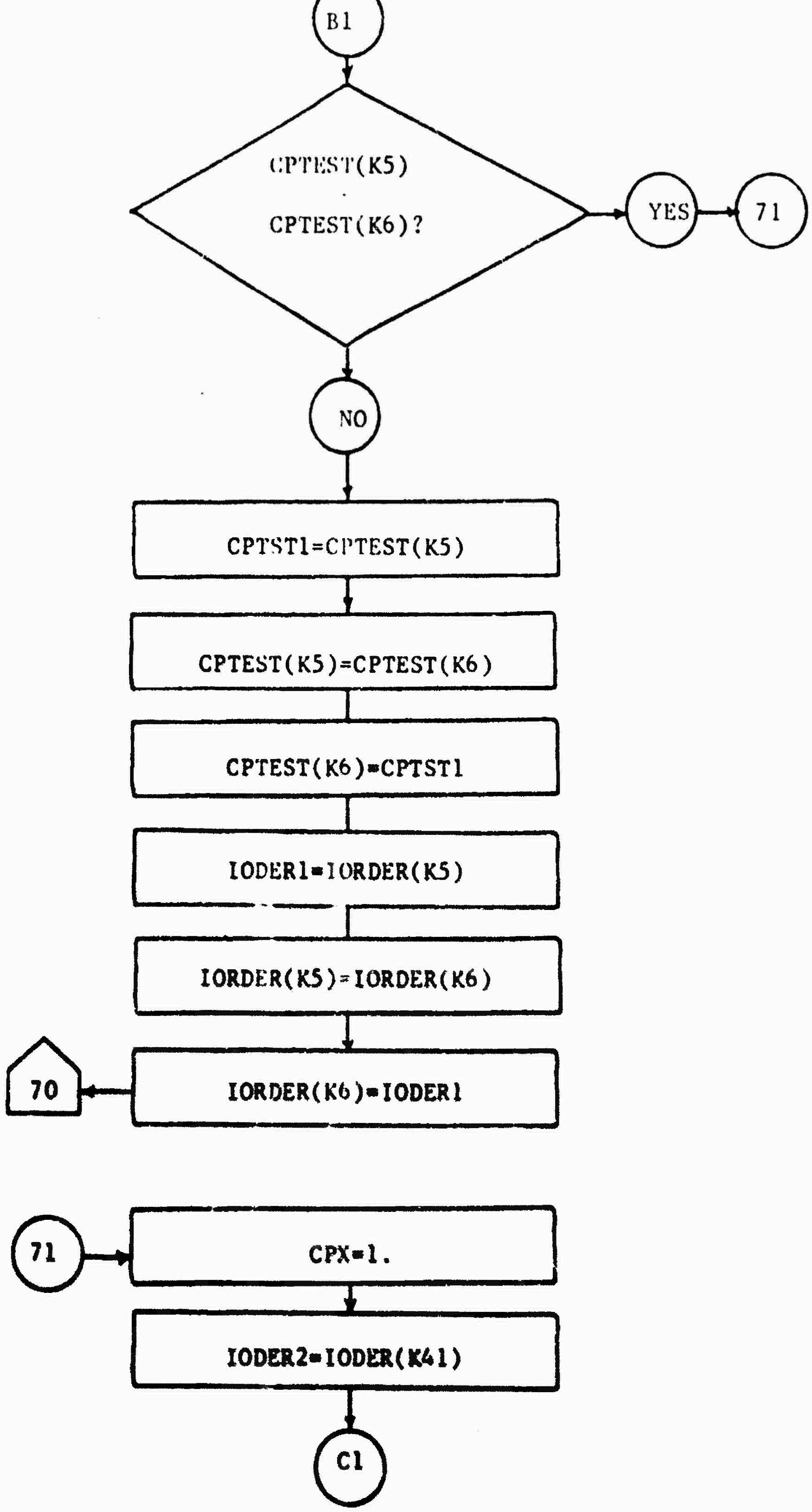

EFFEC
START
COMMON
SAME AS
MAIN
MR=0
MP=0
K40=0
58
K40=K40+1
IC1=IC(J1HOLD,K40)
IORDER(K40)=IC1
K4
<
NASGN
?
YES
58
NO
K43=0
59

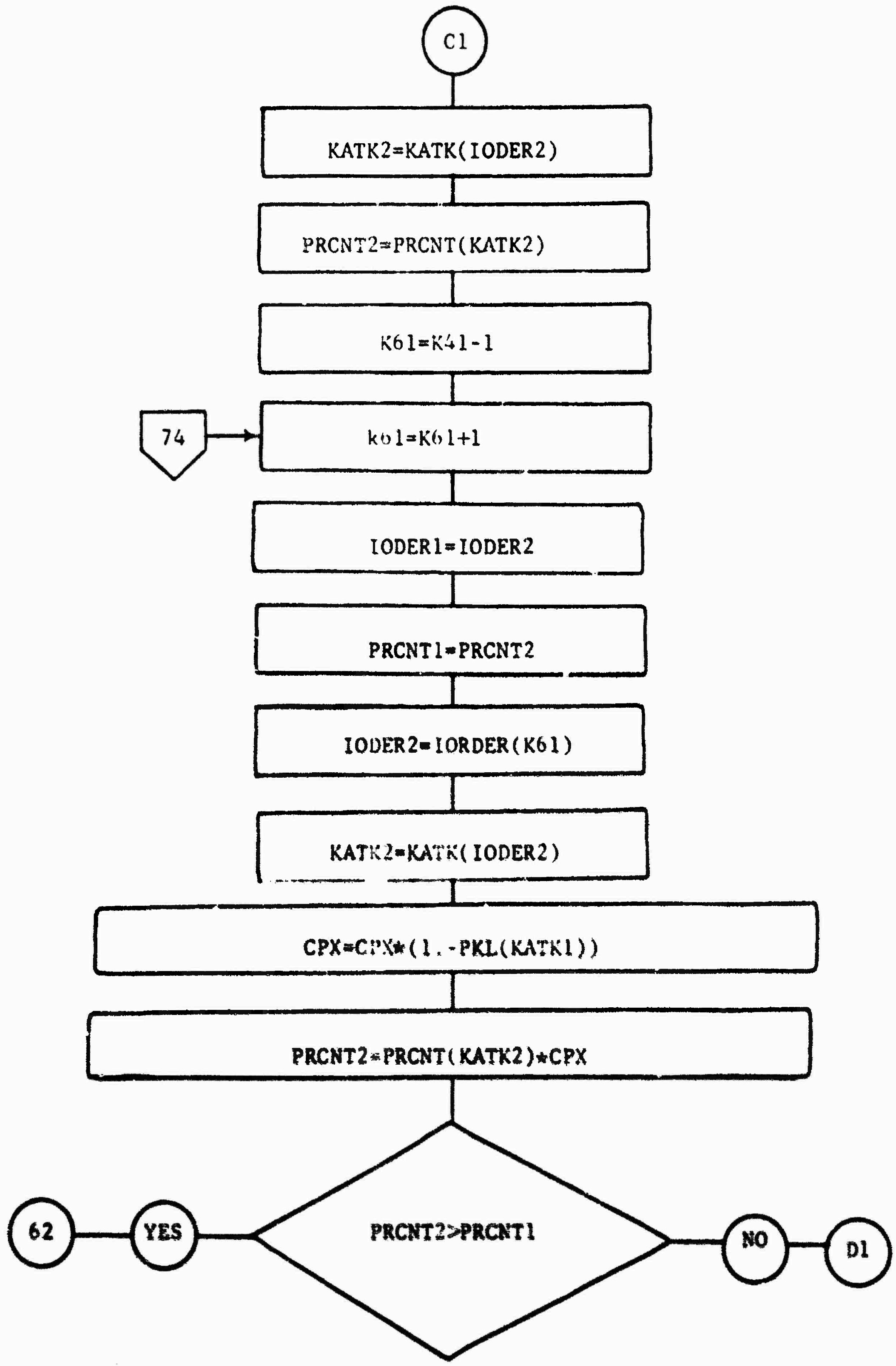

59
K41=K43
MPER=0
MP=0
MPER1=0
61
K41=K41+1
K41
>
NASGN
?
YES
83
NO
K41
NASGN1
?
YES
62
NO
K43=K41
IC1=IORDERNK41)
Y

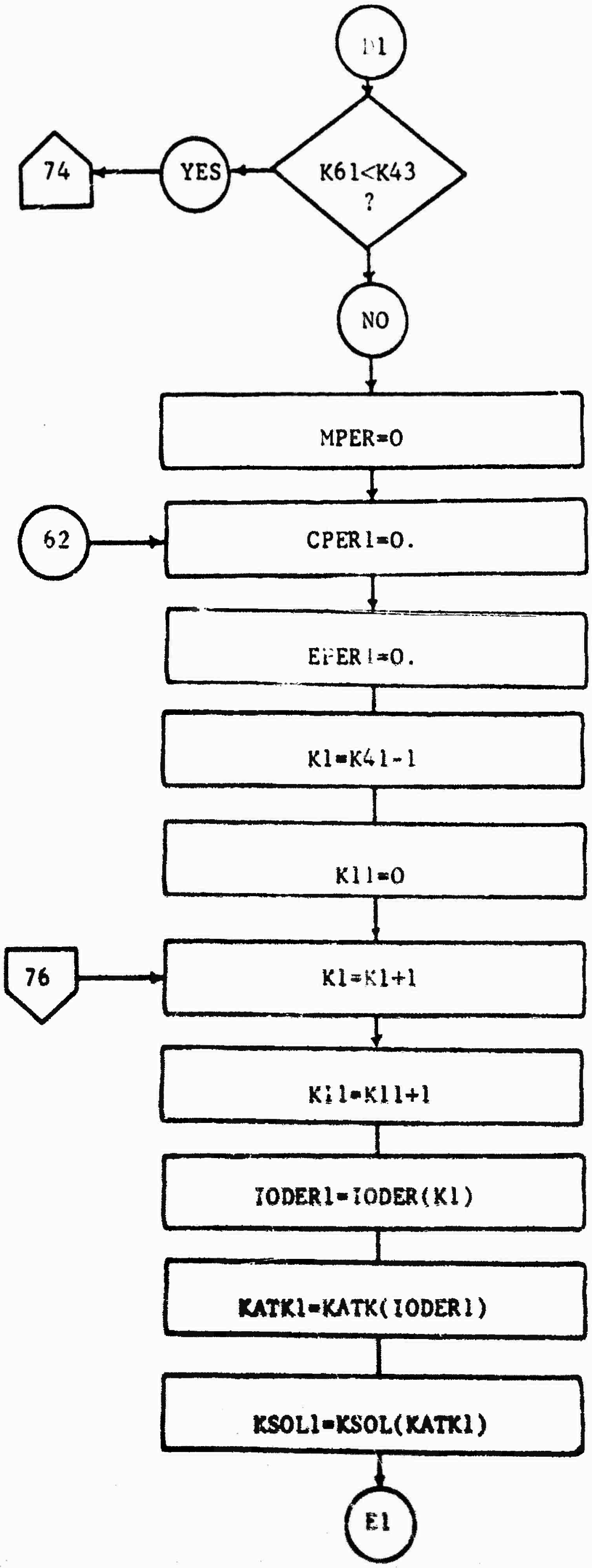

Y
KATK1=KATK(IC1)
K42=K41
63
K42=K42+
IC2=IORDER(K42)
KATK2=KATK(IC2)
JSOL(KATK1) ≠ JSOL(KATK2)?
NO
64
YES
KSOL(KATK1) ≠ KSOL(KATK2)?
NO
64
YES
X

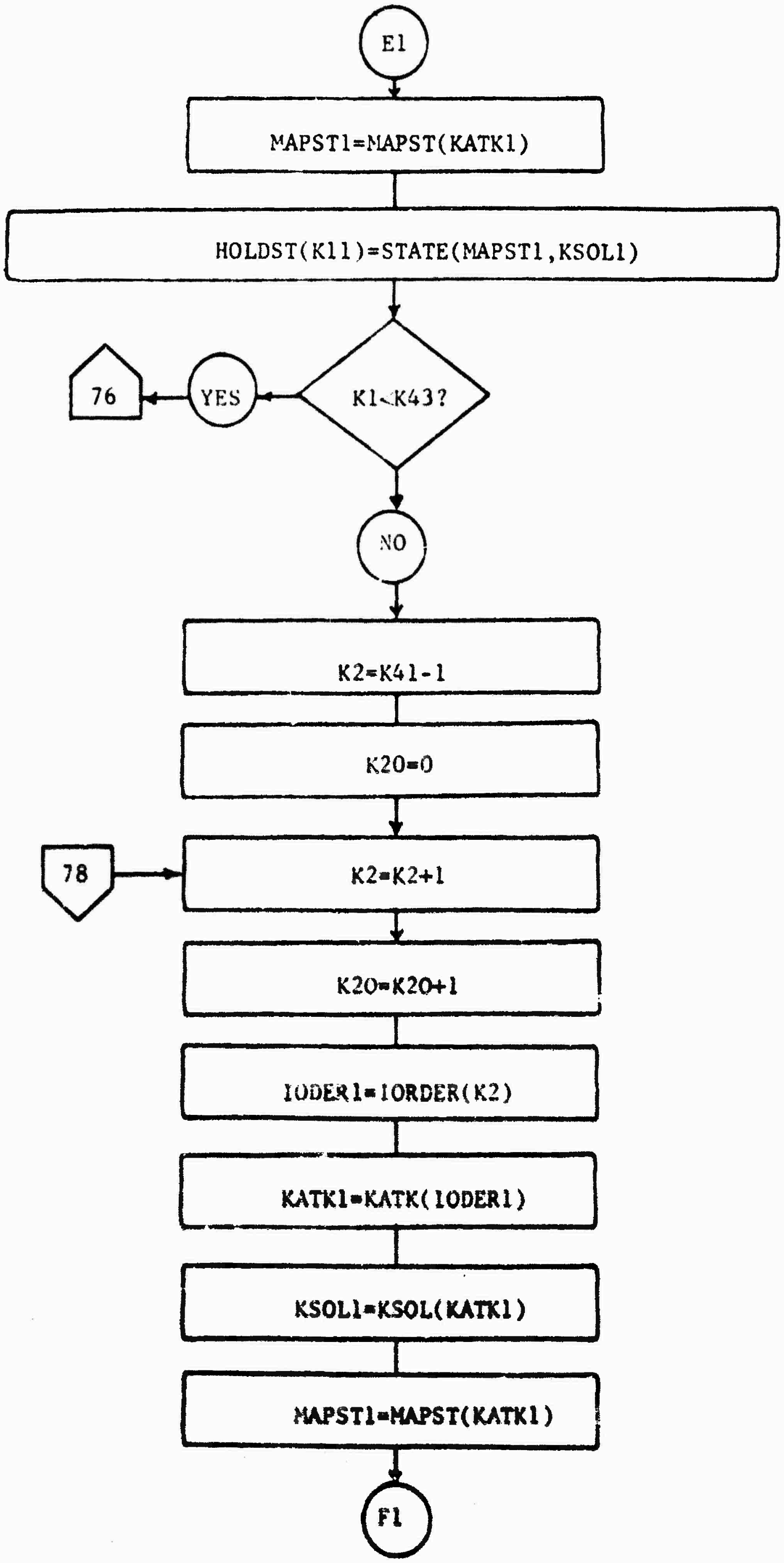

Z
MPER=1
MPER1=1
K43=K43+1
IORDER1=IORDER(K43)
IORDER(K43)=IORDER(K42)
IORDER(K42)=IODER1
K42
NASGN?
YES
63
NO
MPER=0?
YES
62
NO
K4=K41-1
A1

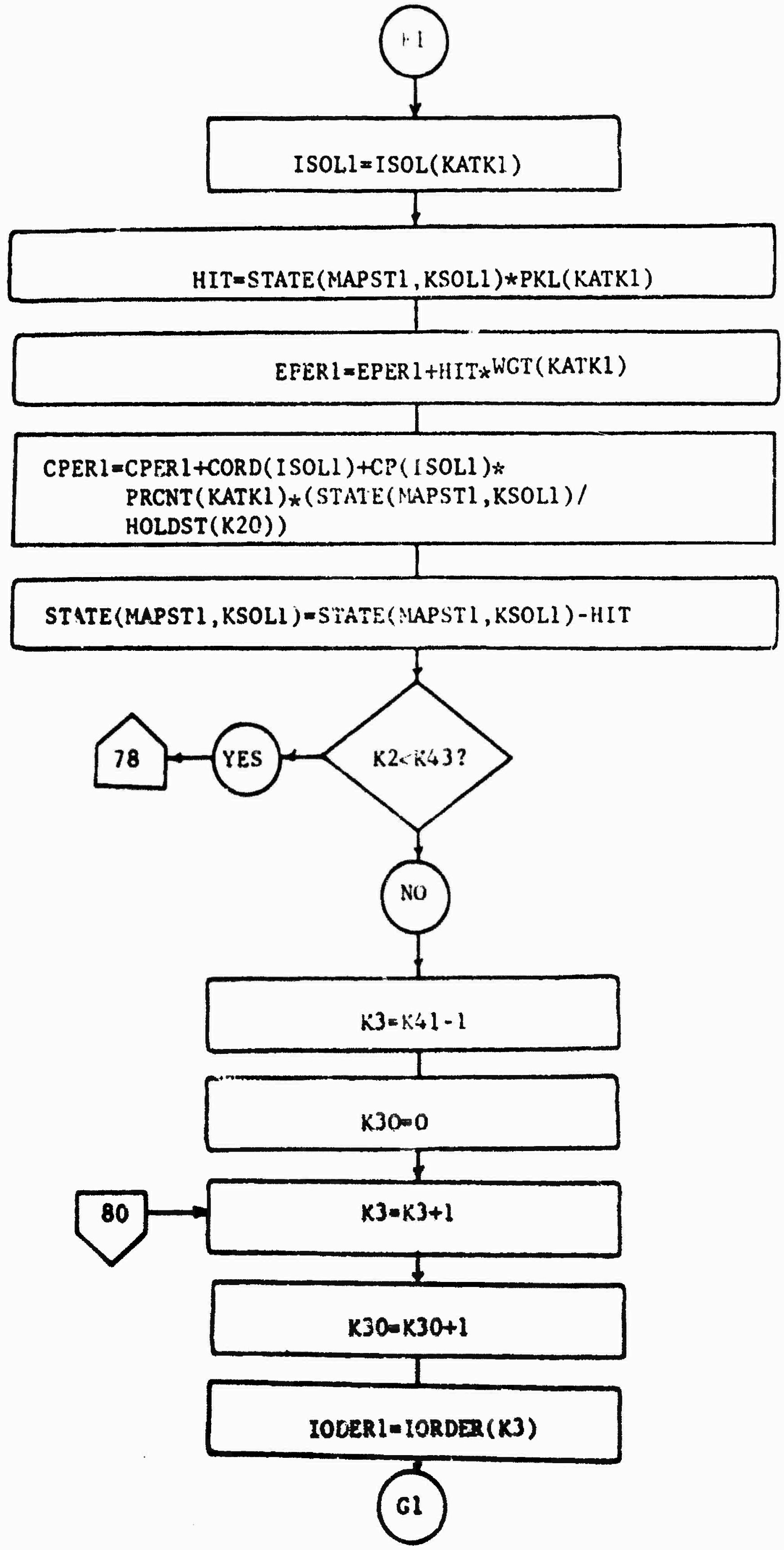

A1
67
K4=K4+1
IODER1=IORDER(K4)
KATK1=KATK(IODER1)
ISOL1=ISOL(KATK1)
CPTEST(K4)=CP(ISOL1)
67
YES
K4<K43?
NO
K5=K41-1
69
K5=K5+1
K6=K5
70
K6=K6+1
71
YES
K6>K43?
NO
B1

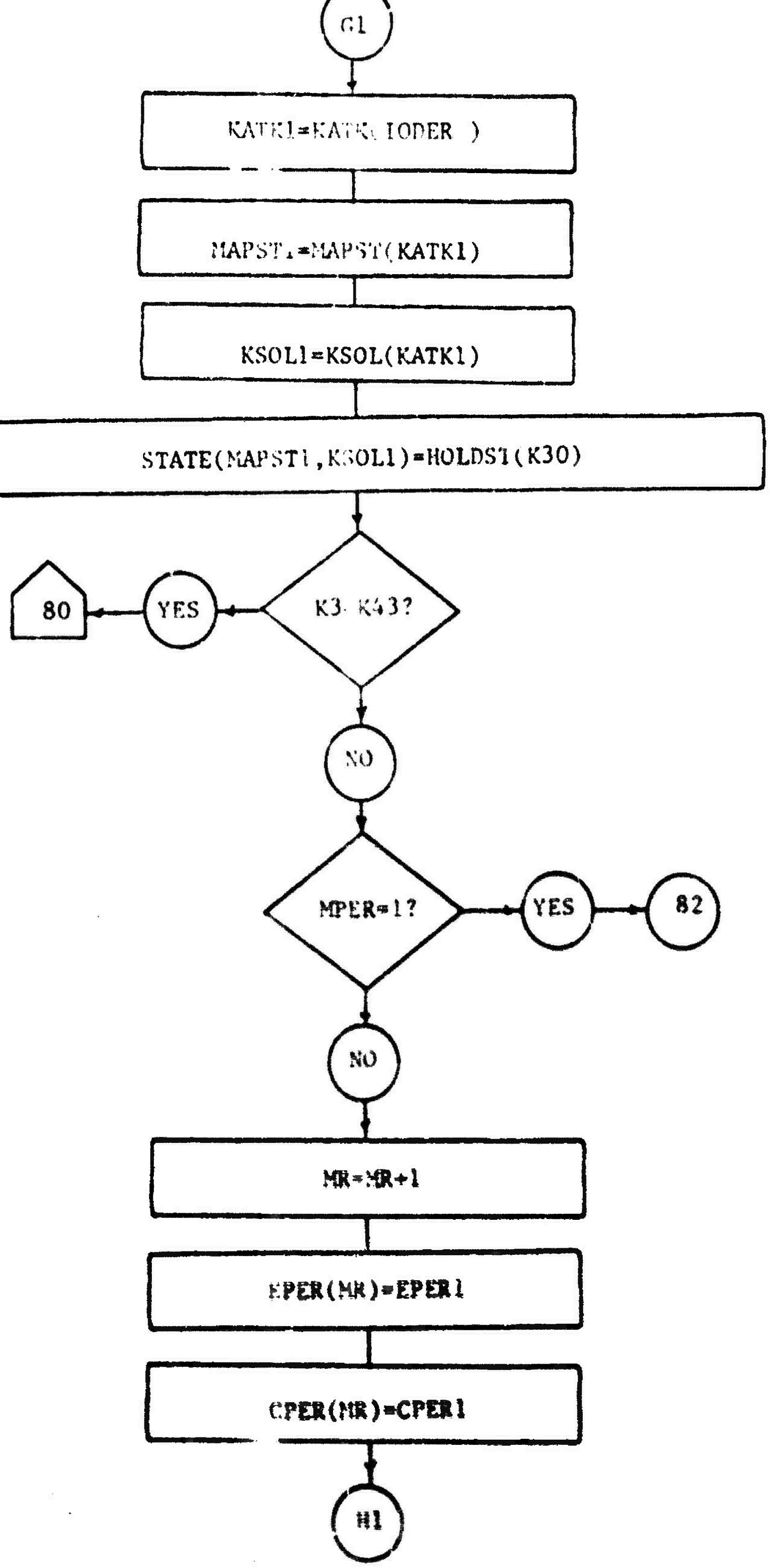

B1
CPTEST(K5)
CPTEST(K6)?
YES
71
NO
CPTST1=CPTEST(K5)
CPTEST(K5)=CPTEST(K6)
CPTEST(K6)=CPTST1
IODER1=IORDER(K5)
IORDER(K5)=IORDER(K6)
IORDER(K6)=IODER1
70
71
CPX=1.
IODER2=IODER(K41)
C1

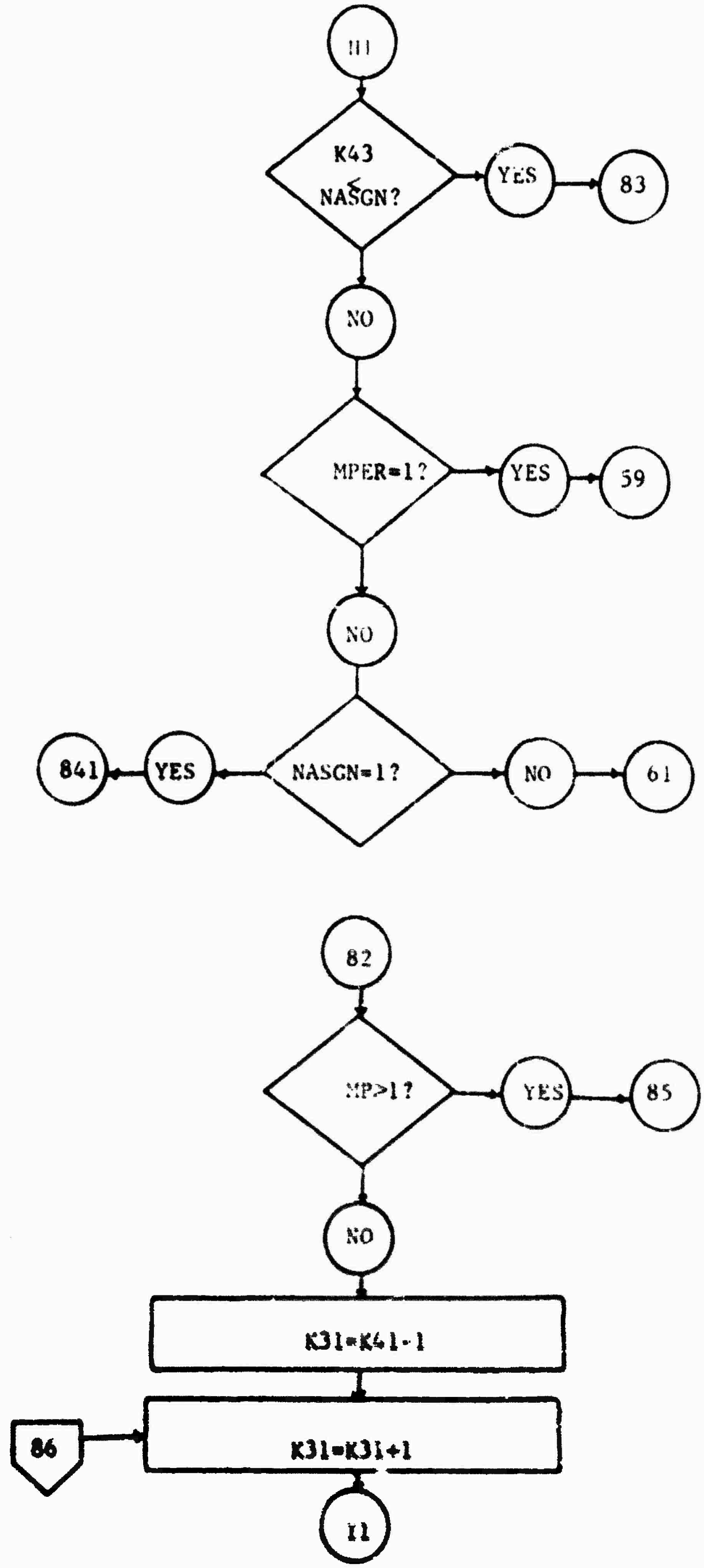

C1
KATK2=KATK(IODER2)
PRCNT2=PRCNT(KATK2)
K61=K41-1
74
k61=K61+1
IODER1=IODER2
PRCNT1=PRCNT2
IODER2=IORDER(K61)
KATK2=KATK(IODER2)
CPX=CPX*(1.-PKL(KATK1))
PRCNT2=PRCNT(KATK2)*CPX
PRCNT2>PRCNT1
62
YES
NO
D1

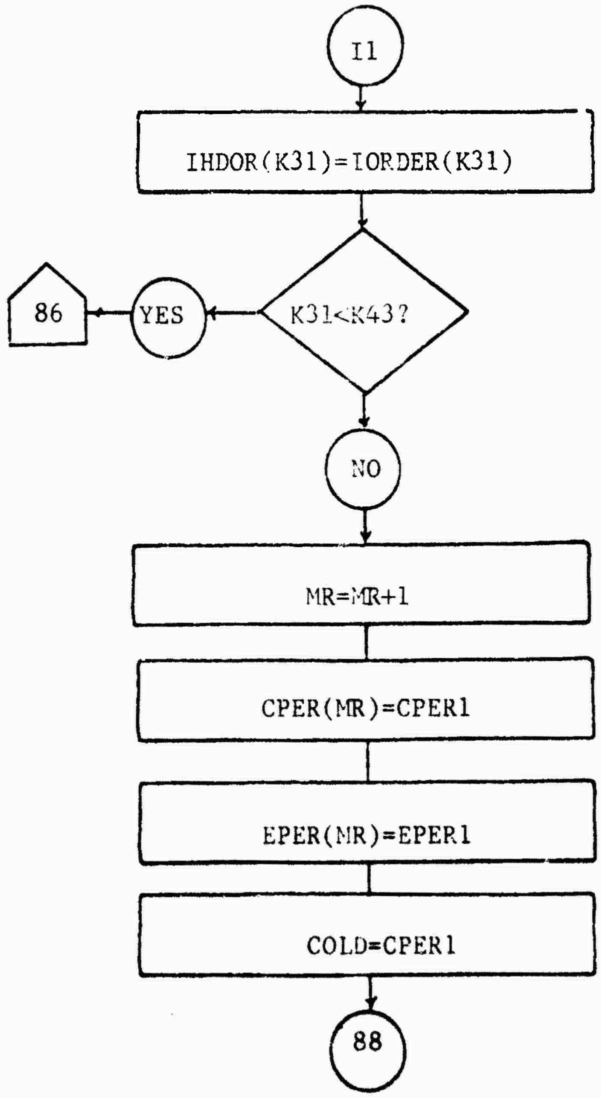

D1
K61<K43 ?
YES
74
NO
MPER=0
62
CPER1=0.
EPER1=0.
K1=K41-1
K11=0
76
K1=K1+1
K11=K11+1
IODER1=IODER(K1)
KATK1=KATK(IODER1)
KSOL1=KSOL(KATK1)
E1

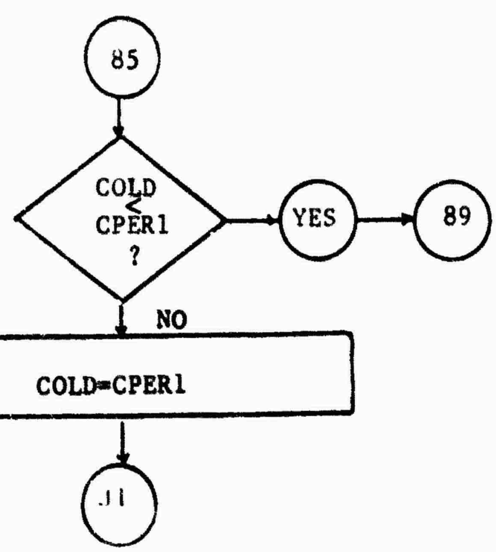

E1
MAPST1=MAPST(KATK1)
HOLDST(K11)=STATE(MAPST1,KSOL1)
K1<K43?
YES
76
NO
K2=K41-1
K20=0
78
K2=K2+1
K20=K20+1
IODER1=IORDER(K2)
KATK1=KATK(IODER1)
KSOL1=KSOL(KATK1)
MAPST1=MAPST(KATK1)
F1

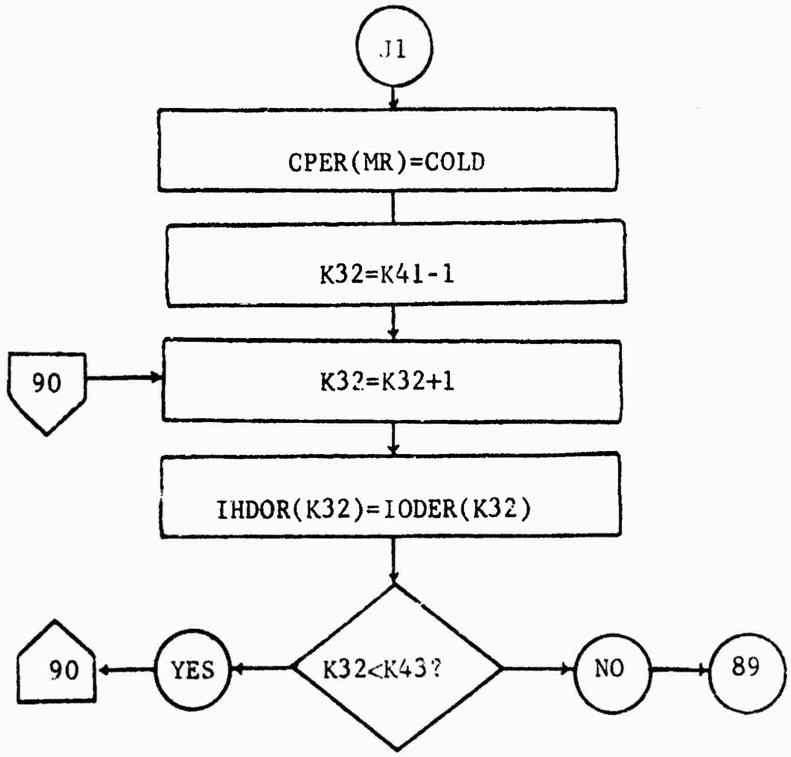

F1
ISOL1=ISOL(KATK1)
HIT=STATE(MAPST1,KSOL1)*PKL(KATK1)
EPER1=EPER1+HIT*WGT(KATK1)
CPER1=CPER1+CORD(ISOL1)+CP(ISOL1)*
PRCNT(KATK1)*(STATE(MAPST1,KSOL1)/
HOLDST(K20))
STATE(MAPST1,KSOL1)=STATE(MAPST1,KSOL1)-HIT
K2<K43?
YES
78
NO
K3=K41-1
K30=0
80
K3=K3+1
K30=K30+1
IODER1=IORDER(K3)
G1

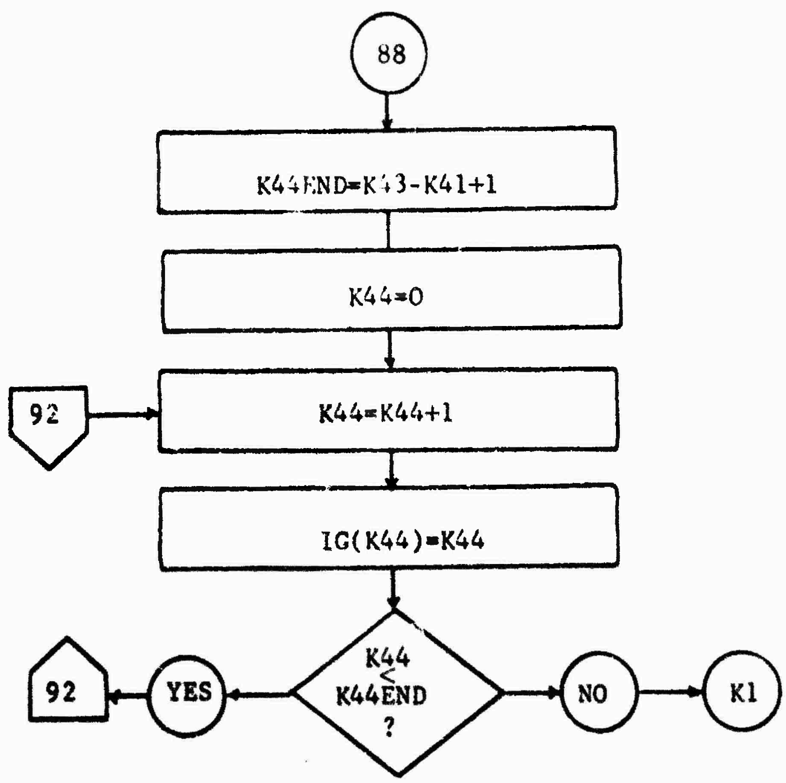

G1
KATK1=KATK(IODER)
MAPST1=MAPST(KATK1)
KSOL1=KSOL(KATK1)
STATE(MAPST1,KSOL1)=HOLDST(K30)
K3 K43?
YES
80
NO
MPER=1?
YES
82
NO
MR=MR+1
EPER(MR)=EPER1
CPER(MR)=CPER1
H1

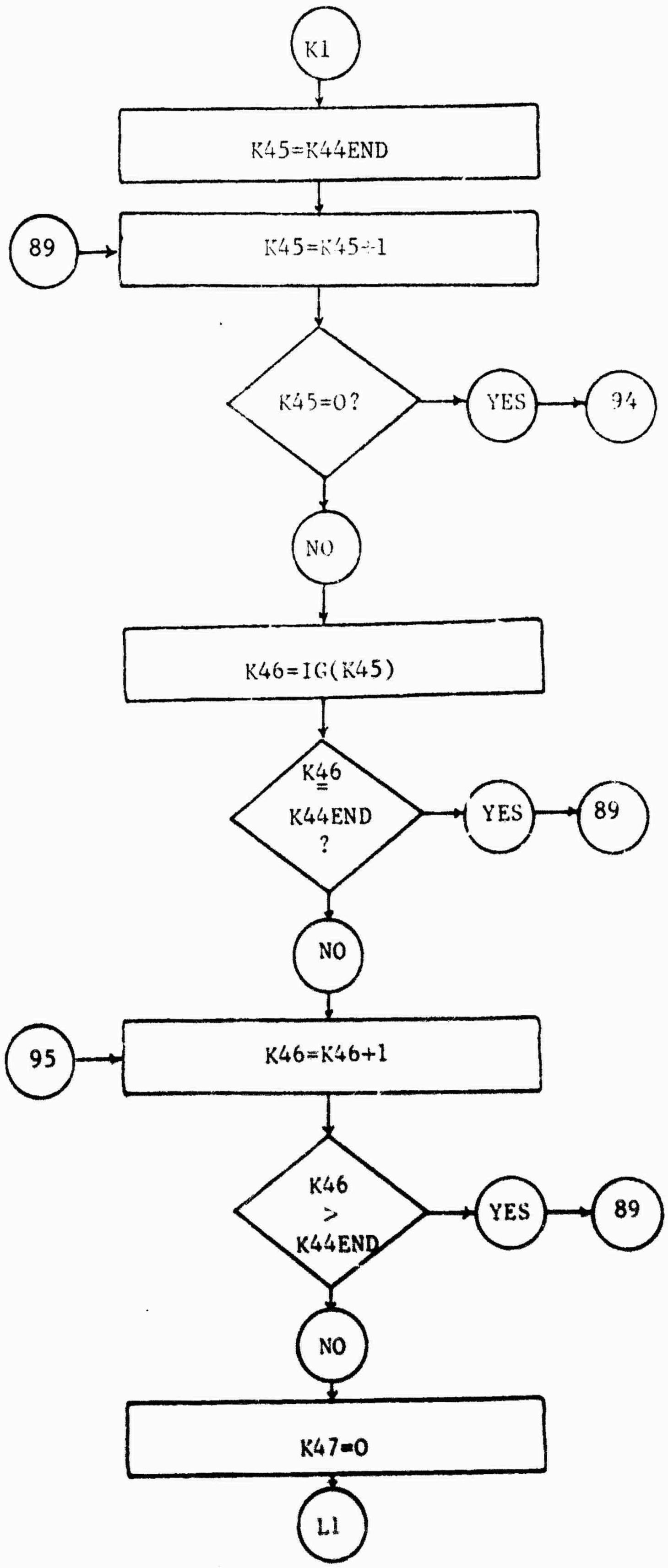

K43
≤
NASGN?
YES
83
NO
MPER=1?
YES
59
NO
841
YES
NASGN=1?
NO
61
82
MP>1?
YES
85
NO
K31=K41-1
86
K31=K31+1
I1

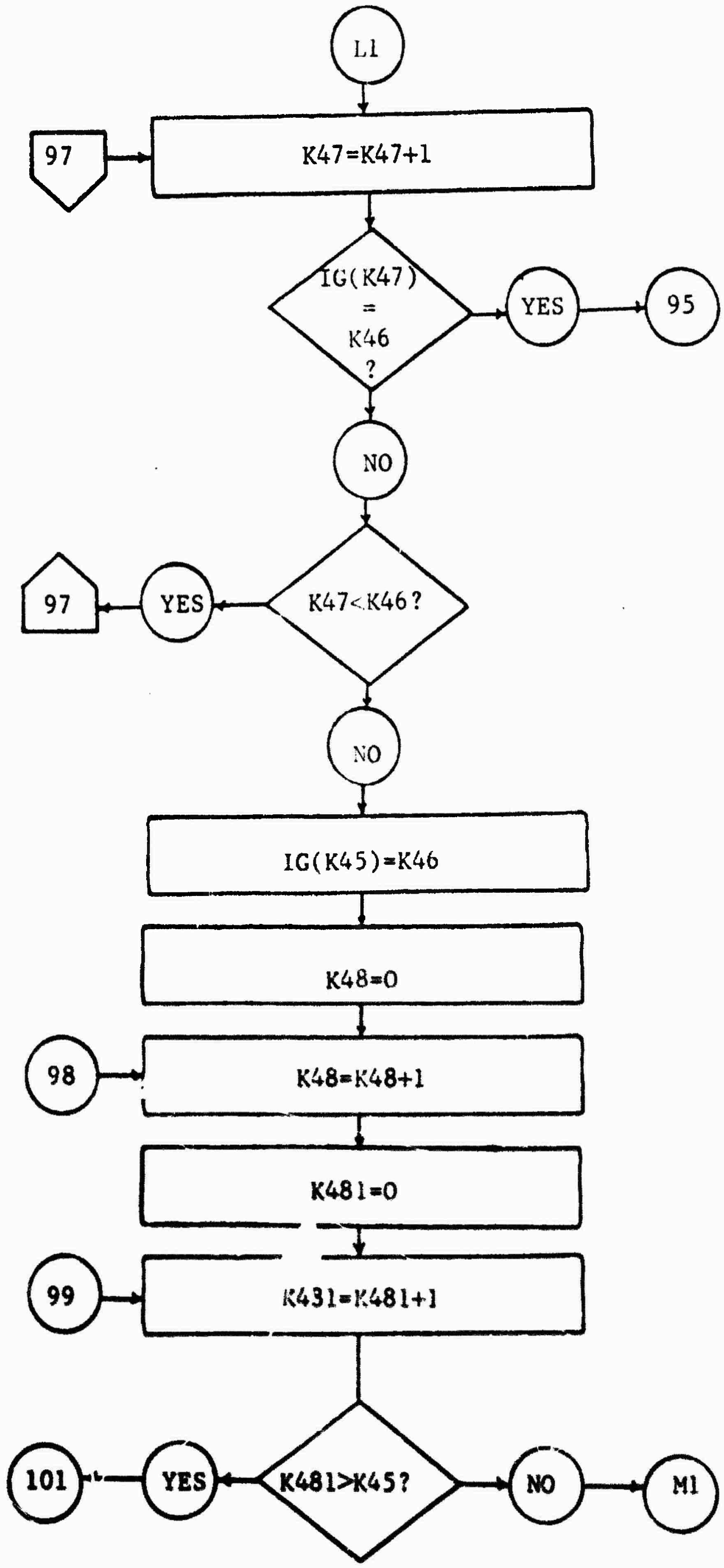

I1
IHDOR(K31)=IORDER(K31)
K31<K43?
YES
86
NO
MR=MR+1
CPER(MR)=CPER1
EPER(MR)=EPER1
COLD=CPER1
88

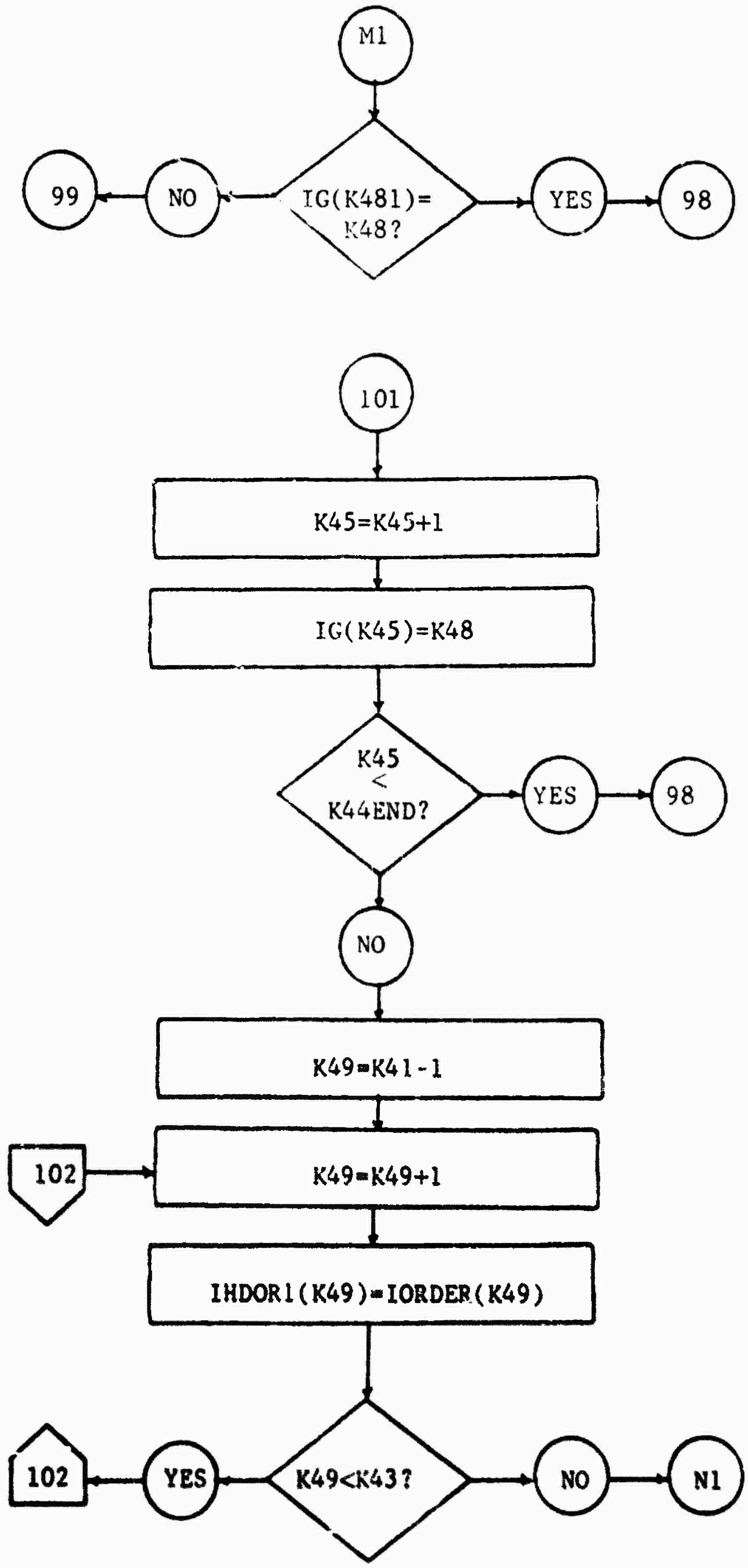

85
COLD
≤
CPER1
?
YES
89
NO
COLD=CPER1
J1


J1
CPER(MR)=COLD
K32=K41-1
90
K32=K32+1
IHDOR(K32)=IODER(K32)
90
YES
K32<K43?
NO
89

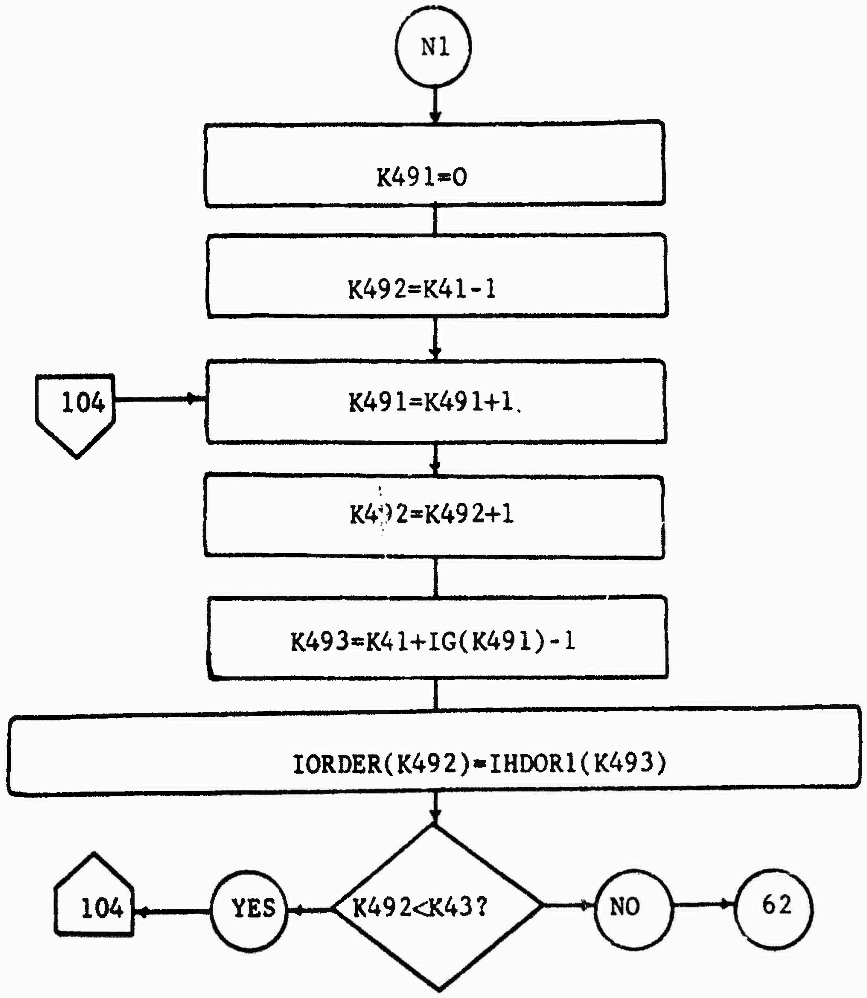

88
K44END=K43-K41+1
K44=0
92
K44=K44+1
IG(K44)=K44
92
YES
K44 < K44END ?
NO
K1

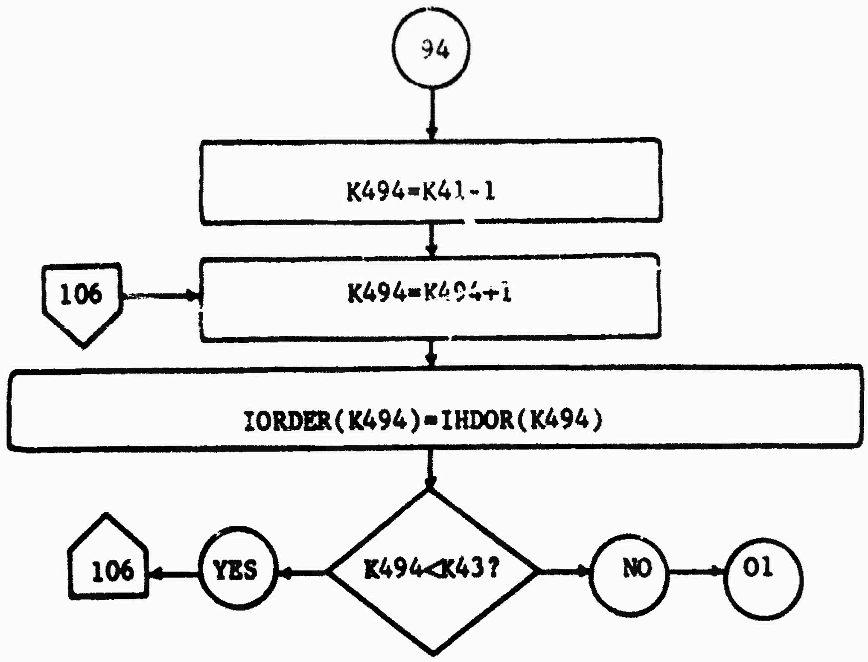

K1
K45=K44END
89
K45=K45+1
K45=0?
YES
94
NO
K46=IG(K45)
K46
=
K44END
?
YES
89
NO
95
K46=K46+1
K46
>
K44END
YES
89
NO
K47=0
L1

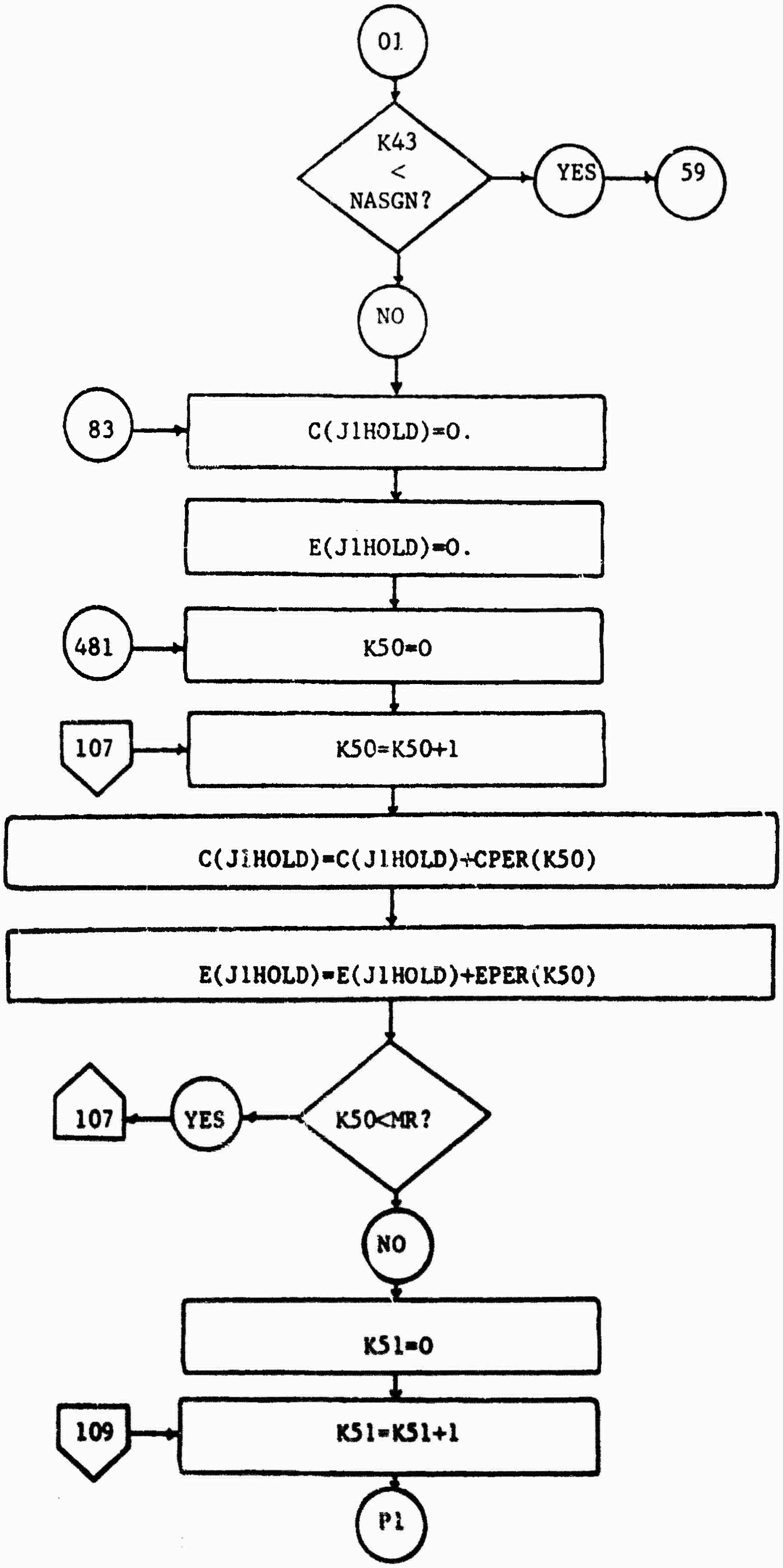

L1
97
K47=K47+1
IG(K47) = K46 ?
YES
95
NO
97
YES
K47<K46?
NO
IG(K45)=K46
K48=0
98
K48=K48+1
K481=0
99
K431=K481+1
101
YES
K481>K45?
NO
M1

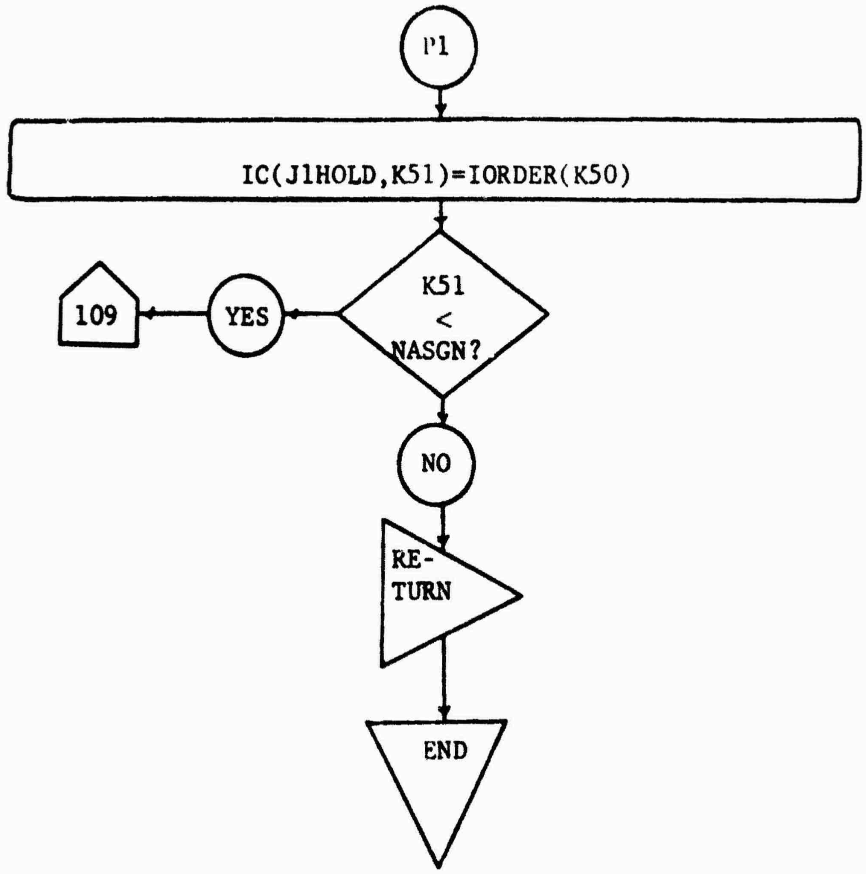


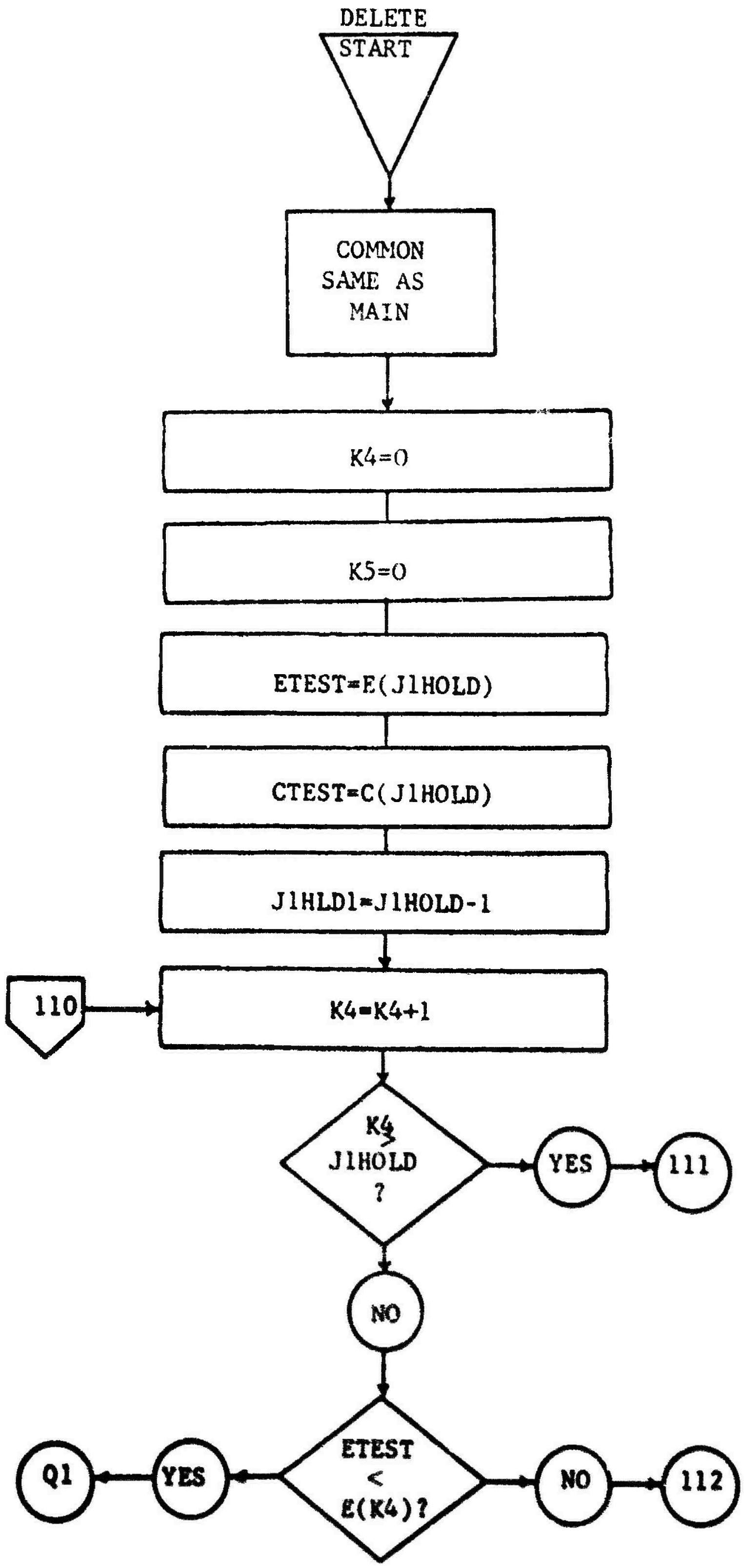
M1
IG(K481)=
K48?
NO
99
YES
98
101
K45=K45+1
IG(K45)=K48
K45
<
K44END?
YES
98
NO
K49=K41-1
102
K49=K49+1
IHDOR1(K49)=IORDER(K49)
K49<K43?
YES
102
NO
N1

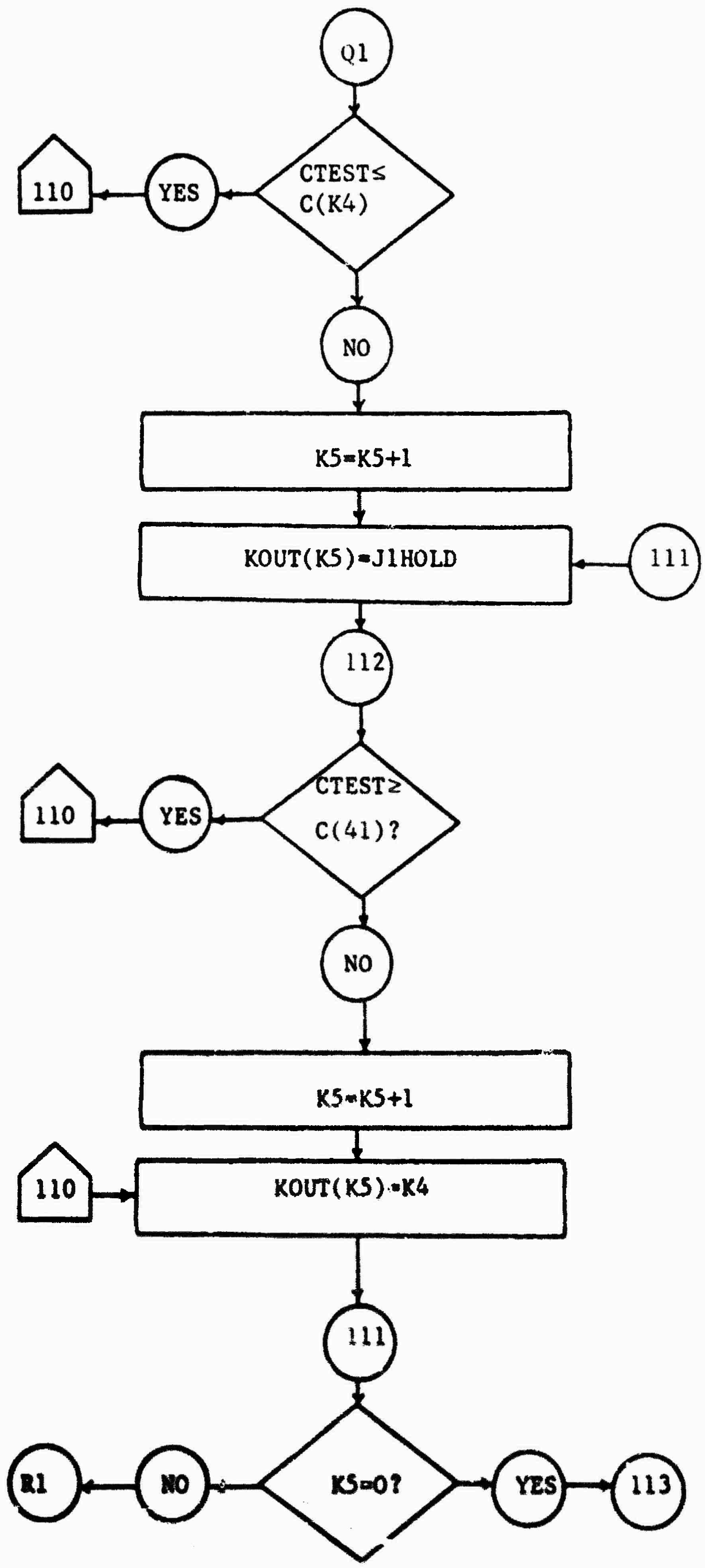

N1
K491=0
K492=K41-1
104
K491=K491+1.
K4?2=K492+1
K493=K41+IG(K491)-1
IORDER(K492)=IHDOR1(K493)
104
YES
K492<K43?
NO
62


94
K494=K41-1
106
K494=K494+1
IORDER(K494)=IHDOR(K494)
106
YES
K494<K43?
NO
01

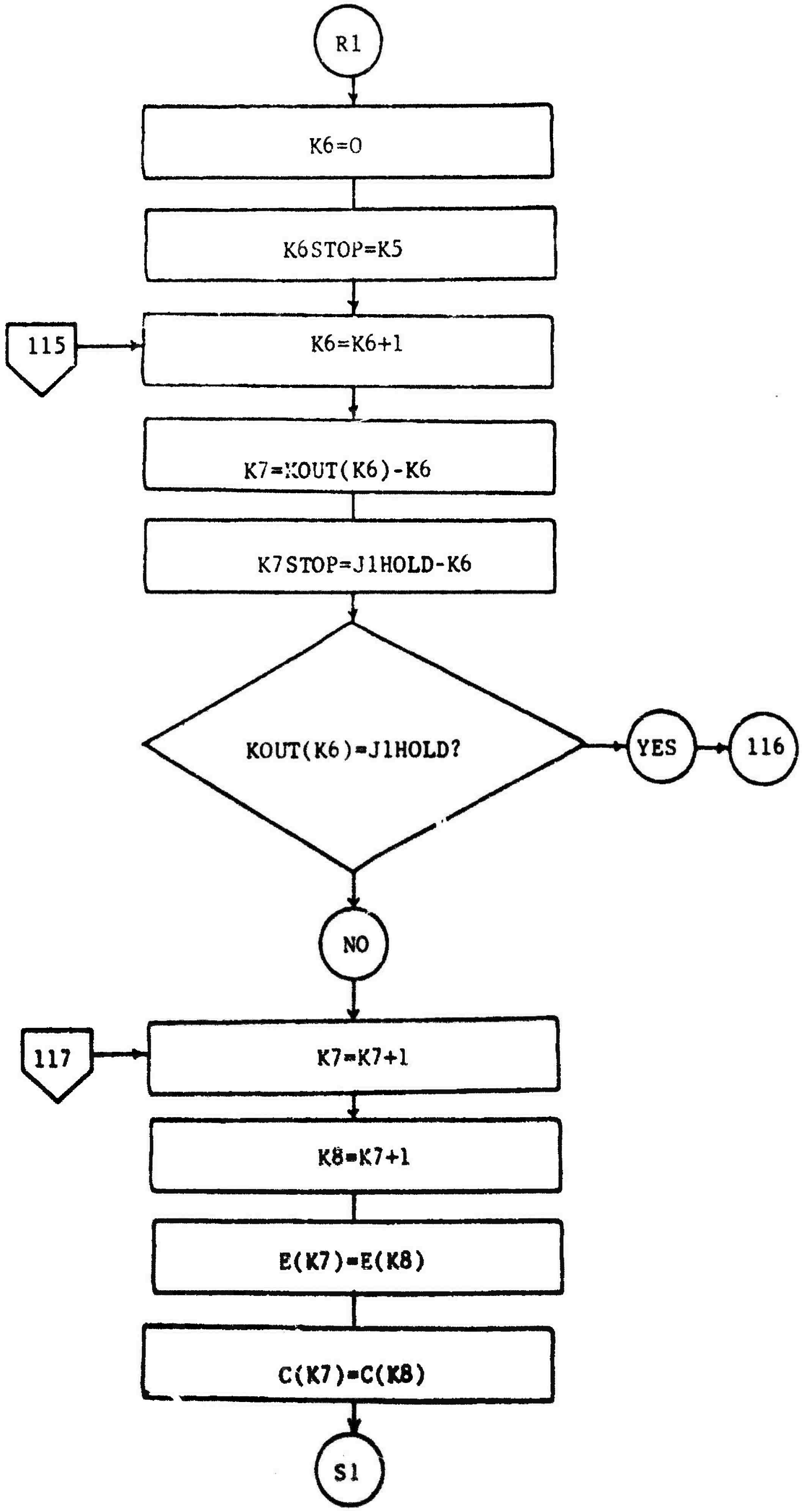

01
K43
<
NASGN?
YES
59
NO
83
C(J1HOLD)=0.
E(J1HOLD)=0.
481
K50=0
107
K50=K50+1
C(J1HOLD)=C(J1HOLD)+CPER(K50)
E(J1HOLD)=E(J1HOLD)+EPER(K50)
107
YES
K50<MR?
NO
K51=0
109
K51=K51+1
P1

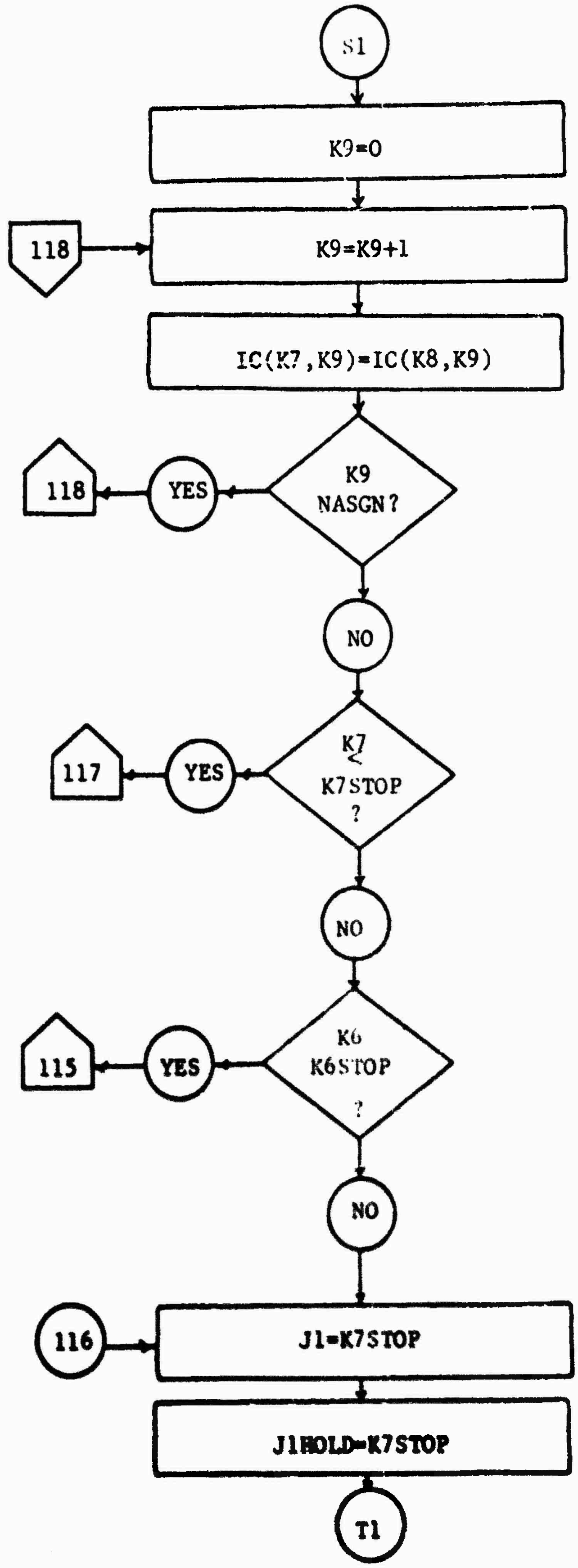

P1
IC(J1HOLD,K51)=IORDER(K50)
K51
<
NASGN?
YES
109
NO
RE-
TURN
END

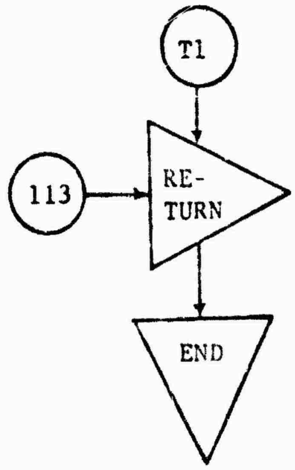

DELETE
START
COMMON SAME AS MAIN
K4=0
K5=0
ETEST=E(J1HOLD)
CTEST=C(J1HOLD)
J1HLD1=J1HOLD-1
110
K4=K4+1
K4 > J1HOLD ?
YES
111
NO
ETEST < E(K4)?
YES
Q1
NO
112

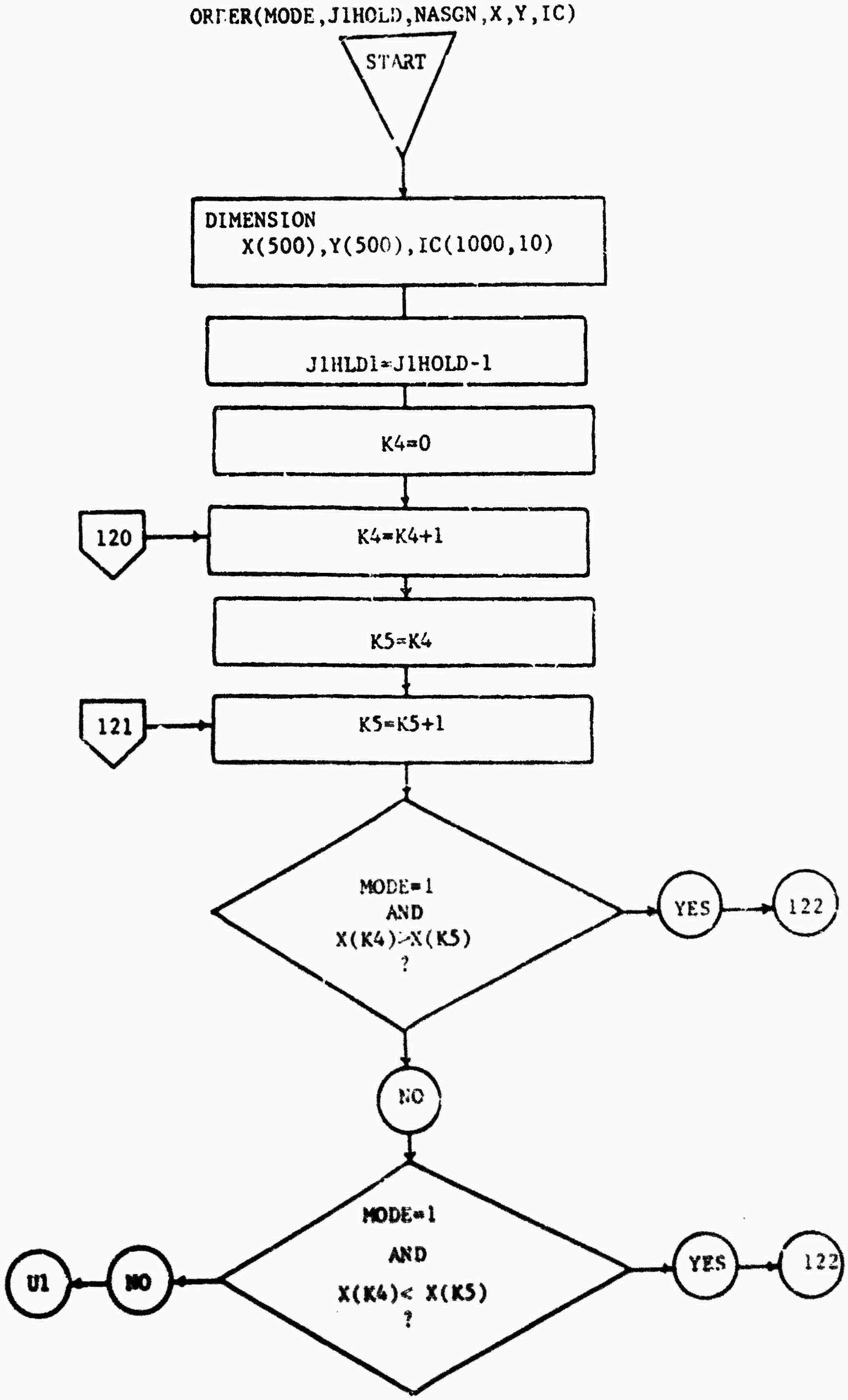


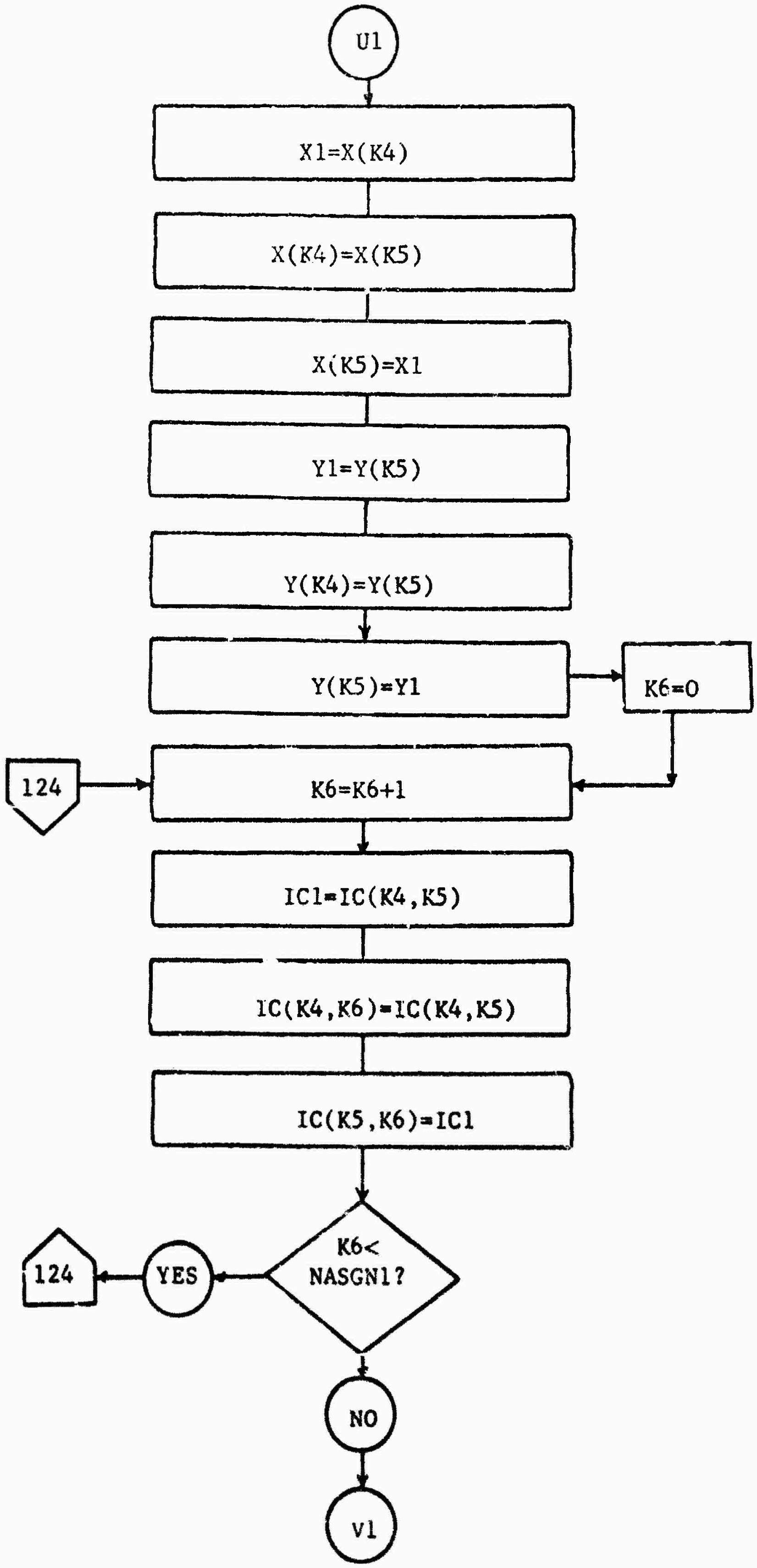
Q1
CTEST≤
C(K4)
YES
110
NO
K5=K5+1
KOUT(K5)=J1HOLD
111
112
CTEST≥
C(41)?
YES
110
NO
K5=K5+1
110
KOUT(K5)=K4
111
K5=0?
NO
R1
YES
113

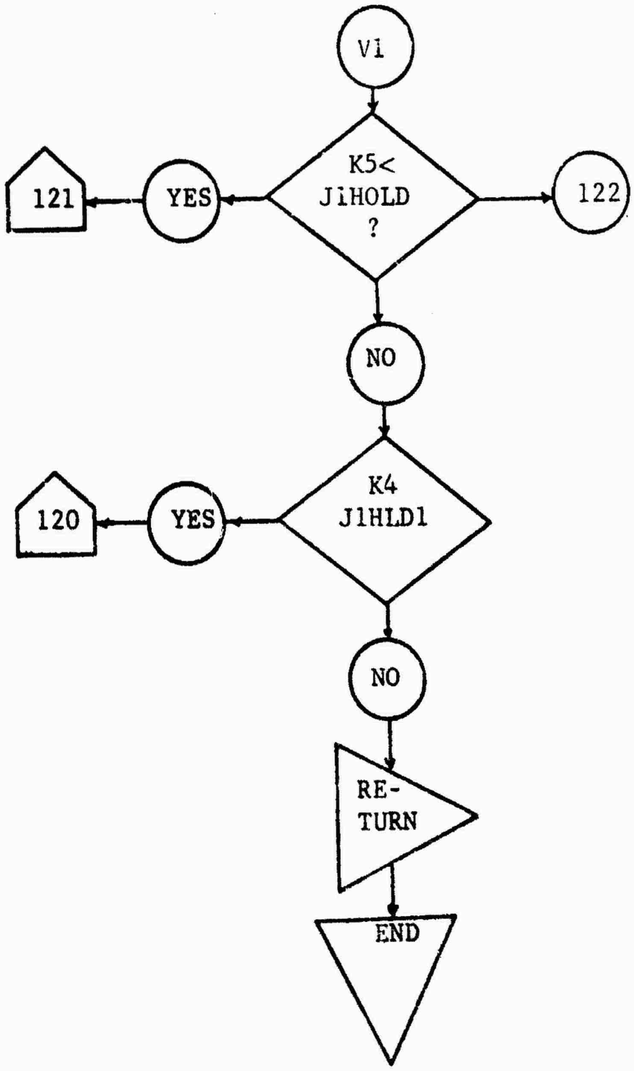

R1
K6=0
K6STOP=K5
115
K6=K6+1
K7=KOUT(K6)-K6
K7STOP=J1HOLD-K6
KOUT(K6)=J1HOLD?
YES
116
NO
117
K7=K7+1
K8=K7+1
E(K7)=E(K8)
C(K7)=C(K8)
S1

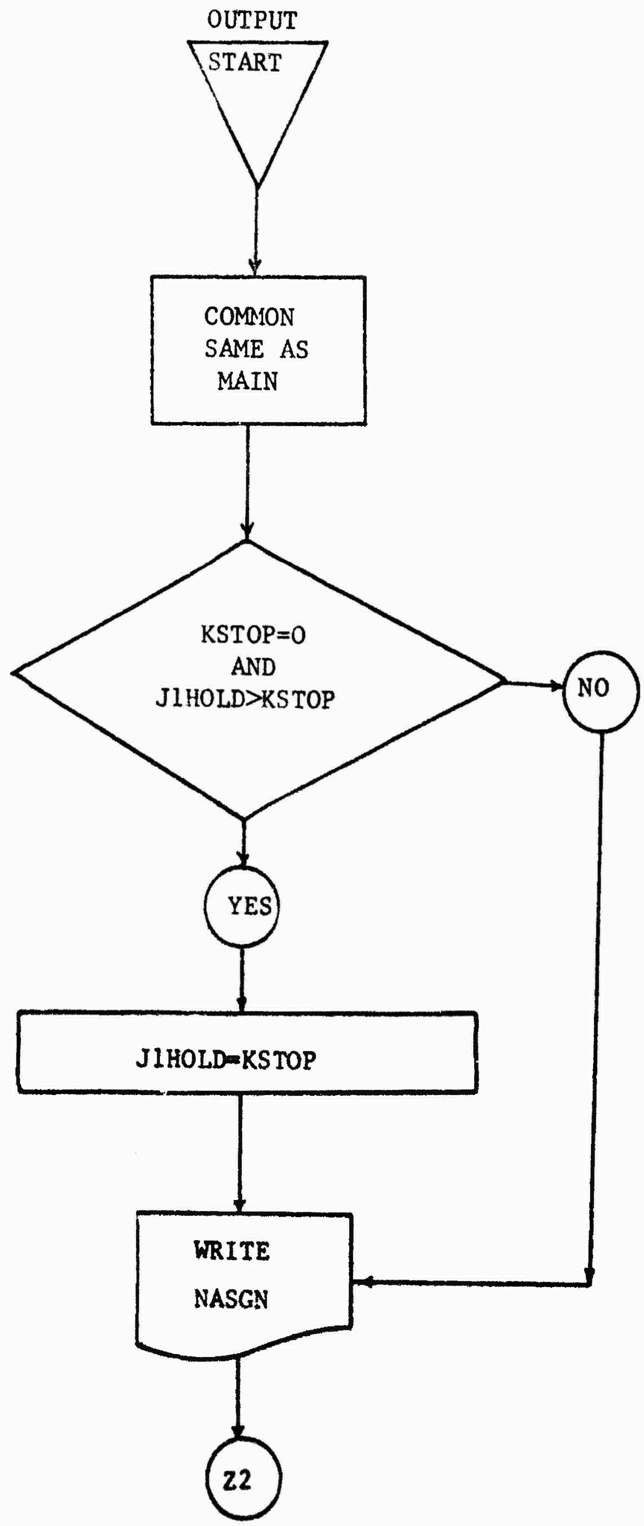

S1
K9=0
118
K9=K9+1
IC(K7,K9)=IC(K8,K9)
K9
NASGN?
YES
118
NO
K7
<
K7STOP
?
YES
117
NO
K6
K6STOP
?
YES
115
NO
116
J1=K7STOP
J1HOLD=K7STOP
T1

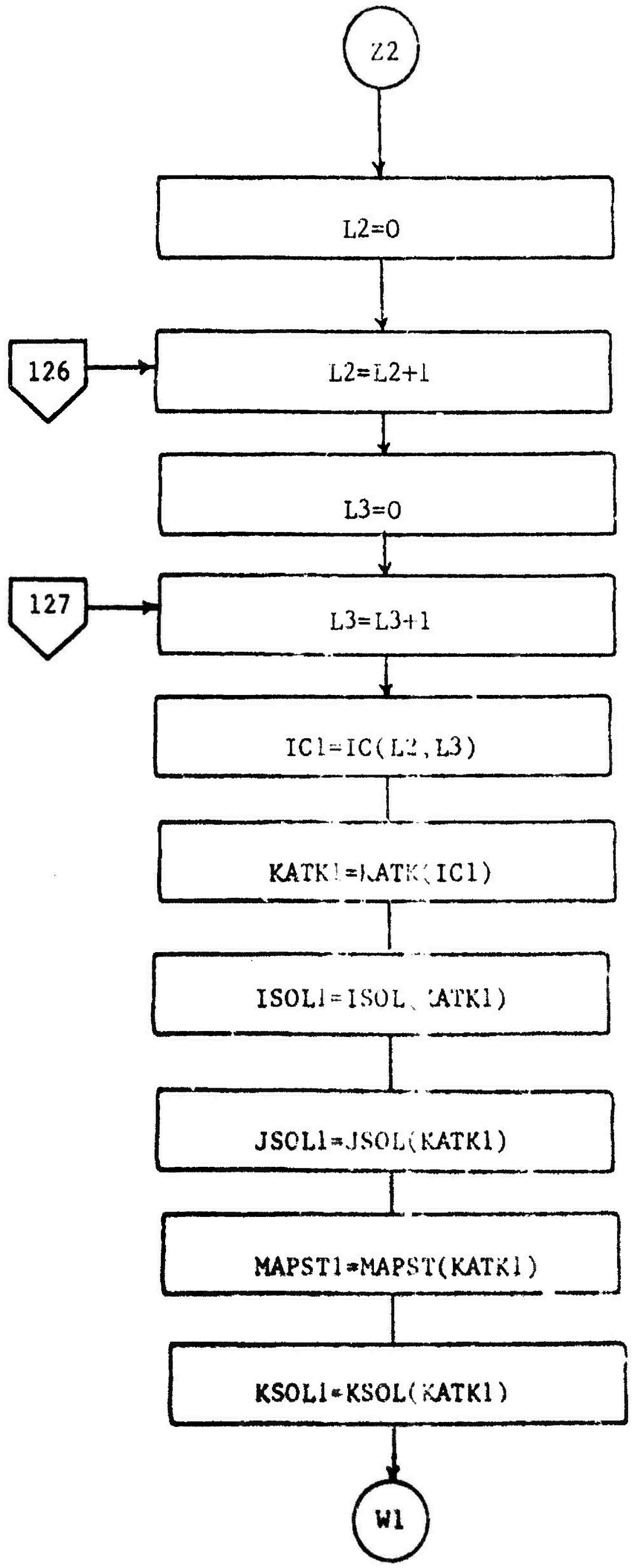

T1
113
RE-
TURN
END

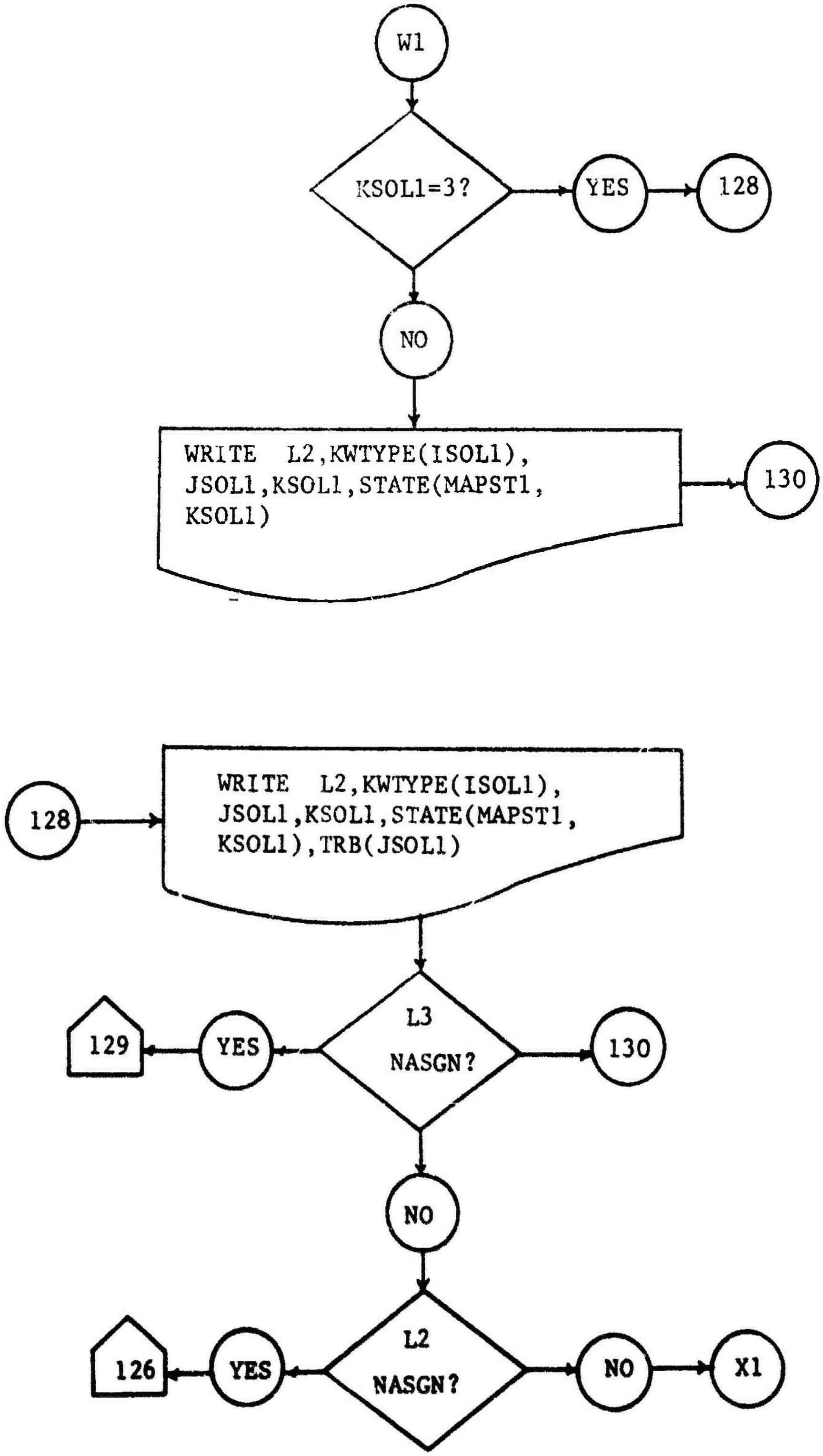

ORDER(MODE,J1HOLD,NASGN,X,Y,IC)
START
DIMENSION X(500),Y(500),IC(1000,10)
J1HLD1=J1HOLD-1
K4=0
120
K4=K4+1
K5=K4
121
K5=K5+1
MODE=1 AND X(K4)>X(K5) ?
YES
122
NO
MODE=1 AND X(K4)< X(K5) ?
YES
122
U1
NO

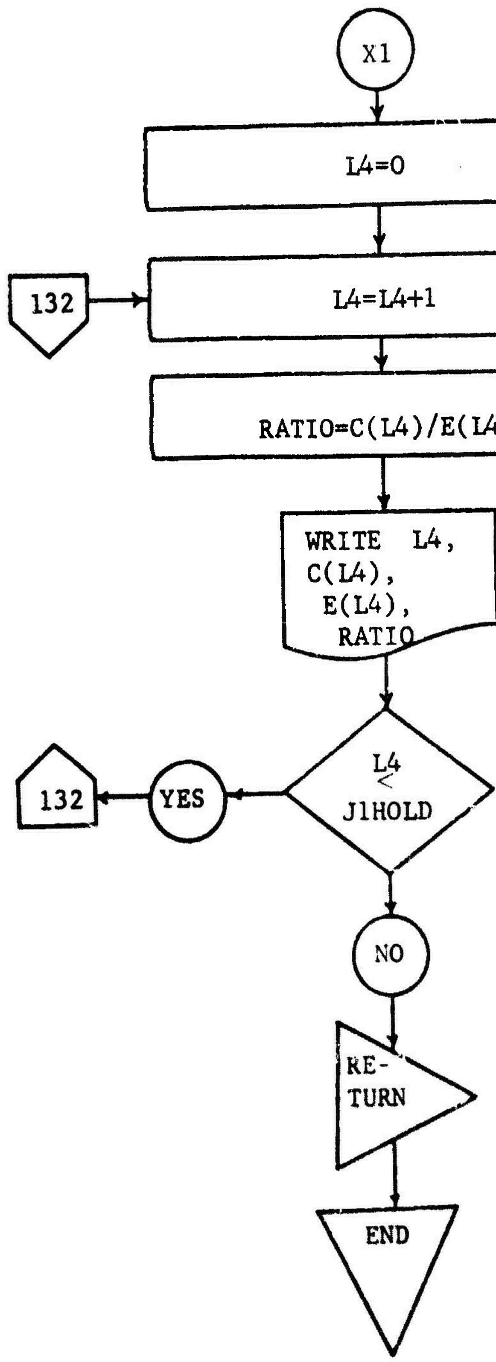

U1
X1=X(K4)
X(K4)=X(K5)
X(K5)=X1
Y1=Y(K5)
Y(K4)=Y(K5)
Y(K5)=Y1
K6=0
124
K6=K6+1
IC1=IC(K4,K5)
IC(K4,K6)=IC(K4,K5)
IC(K5,K6)=IC1
K6<
NASGN1?
YES
124
NO
V1


Vi
K5<
JiHOLD
?
YES
121
122
NO
K4
J1HLD1
YES
120
NO
RE-
TURN
END


OUTPUT
START
COMMON
SAME AS
MAIN
KSTOP=0
AND
J1HOLD>KSTOP
NO
YES
J1HOLD=KSTOP
WRITE
NASGN
Z2


Z2
L2=0
126
L2=L2+1
L3=0
127
L3=L3+1
IC1=IC(L2,L3)
KATK1=KATK(IC1)
ISOL1=ISOL(KATK1)
JSOL1=JSOL(KATK1)
MAPST1=MAPST(KATK1)
KSOL1=KSOL(KATK1)
W1


W1
KSOL1=3?
YES
128
NO
WRITE L2,KWTYPE(ISOL1),
JSOL1,KSOL1,STATE(MAPST1,
KSOL1)
130
128
WRITE L2,KWTYPE(ISOL1),
JSOL1,KSOL1,STATE(MAPST1,
KSOL1),TRB(JSOL1)
L3
NASGN?
YES
129
130
NO
L2
NASGN?
YES
126
NO
X1


X1
L4=0
132
L4=L4+1
RATIO=C(L4)/E(L4)
WRITE L4, C(L4), E(L4), RATIO
L4 ≤ J1HOLD
YES
132
NO
RE-TURN
END

REFERENCES

1. Berg, R. S. 1967. "Armed forces' use of cost-effectiveness analysis." Cost-Effectiveness Analysis: New Approaches in Decision Making. Edited by T. A. Goldman. Frederick A. Praeger, New York.

2. Brunk, H. D. 1965. Mathematical Statistics. Blaisdell Publishing Company, London.

3. English, J. M. 1968. Cost Effectiveness Economic Evaluation of Engineered Systems. John Wiley and Sons, New York.

4. Fields, D. S. 1966. Cost effectiveness analysis: its tasks and their interrelation. J. of O.R., 13:3.

5. Hatry, H. P. 1967. "The use of cost estimates." Cost-Effectiveness Analysis: New Approaches in Decision Making. Edited by T. A. Goldman. Frederick A. Praeger, New York.

6. Heuston, M. C., and G. Ogawa. 1967. "Observations on the theoretical basis of cost-effectiveness." New Approaches in Decision Making. Edited by T. A. Goldman. Frederick A. Praeger, New York.

7. McCullough, J. D. 1967. "Estimating system costs." New Approach to Decision Making. Edited by T. A. Goldman. Frederick A. Praeger, New York.

8. Quade, E. S. 1967. "Introduction and overview." New Approaches in Decision Making. Edited by T. A. Goldman. Frederick A. Praeger, New York.

9. Sieler, K. 1969. Introduction to Systems Cost-Effectiveness. Wiley-Interscience, New York.

10. Wilde, D. J., and C. S. Beightler. 1967. Foundations of Optimization. Prentice-Hall, Englewood Cliffs, N. J.

INITIAL DISTRIBUTION

HQ USAF (SA)	1
AFSC (SDW)	1
AFSC (XRSG)	1
AFSC WEST COAST STDY FAC (XRW)	1
AUL (AUL-LSE-70-239)	1
USAFTFWC (OA)	1
US NAV WPNS CTR (CODE 407)	1
DDC	2
RAND CORP	1
ASD (ENYS)	1
HQ PACAF (IGY)	1
DL	1
DLY	1
DLX	1
DLYE	1
DLR	1
DLYS	1
DLYD	10
DLOSL	2

(The reverse of this page is blank)

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
College of Engineering
Louisiana State University
Baton Rouge, Louisiana 70803

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3. REPORT TITLE
WEAPON SYSTEM EFFECTIVENESS ANALYSIS, OPTIMIZATION AND SIMULATION - PHASE I. VOLUME II. COST-EFFECTIVENESS OF RESOURCE ALLOCATION FOR NETWORK INTERDICTION.

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Final Report, 1 Feb [redacted] 30 Sep [redacted] 70

5. AUTHOR(S) (First name, middle initial, last name)
Lawrence Mann
John Heard Ristrop

(12) 144p.

6. REPORT DATE
Feb [redacted] 71

7a. TOTAL NO. OF PAGES
144

7b. NO. OF REFS
10

8a. CONTRACT OR GRANT NO.
F08635-70-C-0053

b. PROJECT NO.
(16) AF-2543

c. Task No. (17) 254301

d. ~~Work Unit No. 001~~

9a. ORIGINATOR'S REPORT NUMBER(S)

(19)

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
(18) AFATL-TR-71-20 ~~Volume II~~ - V-2

10. DISTRIBUTION STATEMENT
Distribution limited to U. S. Government agencies only; this report documents test and evaluation; distribution limitation applied February 1971 . Other requests for this document must be referred to the Air Force Armament Laboratory (DLYD), Eglin Air Force Base, Florida 32542.

11. SUPPLEMENTARY NOTES
Available in DDC

12. SPONSORING MILITARY ACTIVITY
Air Force Armament Laboratory
Air Force Systems Command
Eglin Air Force Base, Florida 32542

13. ABSTRACT

A cost-effectiveness analysis was performed on the problem of resource allocation for supply network interdiction. Subject to availability constraints Air Force weapon systems are allocated to targets within a supply network. The measures of system performance are cost in dollars and effectiveness in pounds of supplies interdicted. By means of a limited enumerative approach, solutions may be found such that no other alternative assignments of aircraft to targets exhibit simultaneously a higher effectiveness but lower cost than the solutions. Data elements describing the network and weapon system characteristics and weapon system availabilities are processed by a computer model in such a manner as to provide information which will aid a field commander's decision processes.

DD FORM 1 NOV 65 1473

404 858

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Cost-Effectiveness Interdiction Supply Network System Performance Information System Decision Making Aid						